電気通信大学大学院情報理工学研究科

平成 25 年度修士論文

トラヒック特徴量の網羅的評価に基づく未知マルウェアの検知

電気通信大学大学院情報理工学研究科

総合情報学専攻セキュリティ情報学コース

学籍番号　1230017

大月　優輔

主任指導教員　吉浦　裕 教授

市野　将嗣　助教

指導教員　高田　哲司　准教授

提出日　平成 26 年 1 月 30 日（木）

概要

近年，仕事や生活等さまざまな場面でインターネットが必要不可欠な存在となっている．具体的にはインターネットを中心に，イントラネット，モバイルネット，センサーネット，携帯電話網等が重層的に結合し，社会の隅々まで行き届いている．そして，これらのネットワーク上で，行政や金融，流通からコミュニケーション，エンターテインメントに至る多様なサービスが稼働し，人々の生活を支えている．ネットワークがなくては一日も生活できないと言って過言ではない．インターネットが普及し，ユーザの利便性が向上する一方で，それらを悪用した活動の被害が拡大している．これらの活動に用いられる悪意のあるソフトウェア（Malicious Software）の略称からマルウェアと呼ばれている．マルウェアに感染すると，感染した PC のデータ破壊や乗っ取り，個人情報の流出等の被害を受ける可能性があるため，我々の日常生活を脅かす存在となっている．新種（未知）のマルウェア発生数は，2012 年で一日あたり 20 万個（年間 7,300 万），2013 年においては一日あたり 31 万 5000 個（年間約 1.15 億）発生し，依然として増加している．

これに対し，コード等のパターンに基づいてマルウェアの検知，駆除を行うマルウェア対策ソフトがセキュリティベンダによって開発されている．しかしこれらのマルウェア対策ソフトの検知手法は，あらかじめ既知のマルウェアを事前に解析しなければならないため，短期間で大量に出現する未知のマルウェアには対応できない．そのため，未知のマルウェアに感染してしまうことを前提として，感染後にマルウェアの動作だけで早期に検知できる感染検知が必要である．本研究では，ペイロード情報に基づいて感染検知を行う既存研究を分析した．その結果既存研究では，検知手法の検討がメインで，どの特徴量が正常と感染を区別しやすいかという観点からの検討が十分に行われていないことがわかった．個々の特徴量の有効性を明確にできれば，複数の有効な特徴量を組み合わせることで，より正確に正常と感染を識別できる可能性があると考えられる．

そこで本研究では，感染時通信として CCCDATAset，D3M2012 を，正常時通信として 2 種類のイントラネットのトラヒックデータを用いた評価実験を行った．また，マルウェアは，種類毎に異なった通信を行うので，マルウェアを 3 種類（ワーム，トロイの木馬，ファイル感染型ウイルス）に分類した．そして，ペイロードの個々の特徴量について，マルウェアの種類毎の検知の有効性を識別指標 True Positive Rate （以下 TPR）・True Negative Rate （以下 TNR）に基づき，明らかにした．具体的には，261 個の特徴，3 種類 14 検体のマルウェアを用いた実例分析により，ワームで 3 個，トロイの木馬で 15 個，ウイルスで 5 個の，正常/感染かを区別しやすい有効な通信パターンの特徴量（計 20 個）を示す事ができた．

また，ワームでは「インターネット接続確認」等，トロイの木馬では「攻撃通信を行うためのマルウェアのダウンロード」等，ファイル感染型ウイルスでは「IRC 接続」等，の感染活動に，マルウェアの種類毎でペイロード情報に特定の文字列を含んでいる事が確認でき，有効だと判断された特徴量との有効性を確認できた．

さらに，有効だと判断した 20 個の有効な特徴量を用い，2 つの特徴量の組み合わせ，3 つの特徴量の組み合わせにおいて，最適な組み合わせを評価した．その結果，2 つの特徴量の組み合わせでは，正常時通信と感染時通信を分類する境界線において 3 つ（ASCII 文字コード「e」+「o」（「e」: 99.0% → 100%，「o」: 99.0% → 100%）など），感染時通信を正常時通信と誤検知した感染時通信を分類する境界線において 17 つ（ASCII 文字コード「0」+「i」（「0」: 97.5% → 99.4%，「i」: 98.5% → 99.4%）など）の組み合わせが，7 章で有効だと判断した特徴量の識別率よりも，識別率が 2 つとも高くなった．また，3 つの特徴量の組み合わせでは，感染時通信を正常時通信と誤検知した感染時通信を分類する境界線において，7 つ（ASCII 文字コード「e」+「i」+「o」（「e」: 99.0% → 99.5%，「i」: 98.5% → 99.5%，「o」: 97.0% → 99.5%））の組み合わせが，7 章で有効だと判断した特徴量の識別率よりも，識別率が 3 つとも高くなった．これらを用いることで，より正確に正常と感染を識別できる可能性があることを示した．

目次

第1章　序論

1.1　研究背景

近年，仕事や生活等さまざまな場面でインターネットが必要不可欠な存在となっている．具体的にはインターネットを中心に，イントラネット，モバイルネット，センサーネット，携帯電話網等が重層的に結合し，社会の隅々まで行き届いている．そして，これらのネットワーク上で，行政や金融，流通からコミュニケーション，エンターテインメントに至る多様なサービスが稼働し，人々の生活を支えている．ネットワークがなくては一日も生活できないと言って過言ではない．

インターネットが普及し，ユーザの利便性が向上する一方で，それらを悪用した活動の被害が拡大している．これらの活動に用いられる悪意のあるソフトウェア（Malicious Software）の略称からマルウェアと呼ばれている[1]．マルウェアに感染すると，感染した PC のデータ破壊や乗っ取り，個人情報の流出等の被害を受ける可能性があるため，我々の日常生活を脅かす存在となっている．さらに，マルウェア感染の手口は年々複雑化，多様化しており，近年は活動が表面化しにくいボットネットによる被害の増加や Gumblar に代表される Web からの感染も急激に増えている．最近では，スマートフォンの普及に伴い，Android 端末等のモバイル端末に感染する不正プログラムが急増している．Android 端末の他にも，自動車の車載システムのセキュリティを脅かすマルウェアなども存在する[2]．

また，新種（未知）のマルウェアも日々増加している状況である．未知のマルウェア発生数は，2012 年で一日あたり 20 万個（年間 7,300 万），2013 年においては一日あたり 31 万 5000 個（年間約 1.15 億）もの未知のマルウェアが発生し，依然として増加している [3]．これに対し，シグネチャ型などの手法を用いてマルウェアの検知，駆除を行うマルウェア対策ソフトがセキュリティベンダによって開発されている[4]．しかしこれらのマルウェア対策ソフトの検知手法は，マルウェア毎に特徴を示すシグネチャを用意しなければならず，短期間で大量に出現する未知マルウェアに対応できない．そのため，感染してしまうことを前提として，感染後に早期に検知する技術（感染検知）が必要とされている．

マルウェアに感染した PC はマルウェア感染時に特有な通信（トラヒックデータ）を行うとが考えられる．よって，PC の通信に利用されるトラヒックデータに着目して未知のマルウェアを検知する．また，感染検知では主に感染したホストのトラヒックデータに着目している．本研究ではその一手法として，ヘッダよりも情報が多く，感染時通信と正常時通信を区別する情報が多く含まれていると考えられる，ペイロードを用いた検知手法を取り上げる．本研究ではまず，ペイロード情報に基づいて感染検知を行う既存研究を分析した．その結果，検知手法の検討がメインで，どの特徴量が正常と感染を区別しやすいかという観点からの検討が十分に行われていないことがわかった．個々の特徴量の有効性を明確にできれば，複数の有効な特徴量を組み合わせることで，より正確に正常と感染を識別できる可能性があると考えられる．

1.2　研究目的

感染検知において未知のマルウェアの検知技術を確立するため，以下の目的に従って研究を行う．

(1) ペイロードの個々の特徴量の有効性を明らかにする．
(2) ペイロードの有効な特徴量の最適な組み合わせを明らかにする

1.3　研究方法

本研究の目的に従って，以下の研究を行う．

(1) トラヒックデータの特徴量抽出と評価
(2) 感染検知を行う識別器・コードブックの実装
(3) 識別器による判定の妥当性を検討

第2章　マルウェア

2.1　マルウェア

マルウェアとは，悪意のあるソフトウェア（Malicious Software）の略称であり，ユーザの望まない不正な動作を行うプログラムの総称として，2001 年頃から広く使われるようになった．

具体的には，ネットワークを通じてコンピュータに侵入しデータの破損をもたらすウイルスや，他のコンピュータへの感染をもたらすワーム，ユーザのオンライン・バンキング等の個人情報を盗み出すスパイウェア，ユーザが望まない広告を無理やり表示したりするアドウェア等があり，非常に多種多様なものとなっている．さらに，それらのマルウェアはウイルス対策ソフトウェアによる検知・駆除を回避するために，マルウェア自身のファイル構造の暗号化や，OS を改ざんする自己防衛機能がある．また，マルウェアが正常な PC のソフトウェアにセキュリティ上の脆弱性（セキュリティホール）を発見したとき，セキュリティホールの対策が行われる前にその脆弱性を悪用して攻撃（ゼロデイ攻撃）を行う等，多様化かつ巧妙化を続けている．

このような多種多様なマルウェアは，主に 4 つの種類に分類することが出来る[4].

2.1.1　「感染形態」に着目したマルウェア

(1)　ウイルス

ウイルスは，自分自身を他のファイルやプログラムに寄生させる感染形態のマルウェアであり，感染対象によってブートセクタ感染型とファイル感染型の 2 つに大別できる．ブートセクタ感染型は，フロッピーディスクやハードディスク等のシステム領域を感染対象とし，ファイル感染型は，実行可能ファイルを主な感染対象とする．また，ウイルスの中には他のウイルスを感染対象とするものも存在する．

(2)　ワーム

ワームは，宿主となるファイルやプログラムを必要とせず，単体で動作し自己繁殖を行う感染形態のマルウェアである．ワームは感染力が高いため，大規模感染を引き起こす傾向にある．ワームの感染手法には，電子メールやリバーブルメディア（USB メモリ等）を移動媒体とするもの，Windows のファイル共有やメッセージング機能を利用するもの，そして，OS やアプリケーションの脆弱性に対する攻撃コードを用いるもの等が存在する．

(3)　トロイの木馬

トロイの木馬は，有用なプログラムを装ってユーザ自身によるシステムへの導入・起動を誘い，実際にはユーザの意図しない不正な動作を行うマルウェアである．また，自身はユーザの不注意を利用してシステムへの侵入を果たすため，感染機能を持たないものが多い．

2.1.2　「目的」に着目したマルウェア

・スパイウェア

スパイウェアは，ユーザの PC 上で個人情報や行動履歴を収集し，特定のサーバに送信することを目的としたマルウェアである．ユーザのキーボード操作を記録・収集するマルウェアであるキーロガーも，目的という点でスパイウェアの一種と考えられる．

・アドウェア

アドウェアは，ユーザに企業広告等を提示することを目的としたプログラムであり，ユーザの同意なしに広告を頻繁にポップアップしたり，ユーザの意図しない Web サイトに強制誘導したりするマルウェアである．

・ランサムウェア

ランサムウェアは，ユーザの PC 上のディレクトリやファイルに対して，強制的に暗号化やパスワード付き ZIP 圧縮を行うことでユーザのデータを「人質」にし，そのデータの復号や解凍の見返りとして，ユーザから身代金（ransom）を搾取するマルウェアである

・スケアウェア

スケアウェアは，ユーザに虚偽の情報を提示し不安（scare）を煽ることで，無意味なソフトウェアを販売するマルウェアである．例としては，まずスケアウェアは「あなたのコンピュータにはスパイウェアとウイルスが存在します. 今すぐ, この対策ソフトをインストールしてください」といった偽情報をユーザに提示する．その後，偽のウイルス対策ソフトが保管してある Web サイ

トにユーザを誘導し，実際には何の機能も有さないプログラムをウイルス対策ソフトと称してユーザに購入させる．それによって攻撃者は代金やクレジットカード番号を奪い取るものである．

2.1.3 「機能」に着目したマルウェア

・ダウンローダ

ダウンローダは，自身とは別のマルウェアを特定のサイトからダウンロードし，感染 PC にインストールする機能を持つマルウェアである．ダウンローダがあることによって，マルウェアのダウンロード回数が増大し，セキュリティベンダにマルウェア自身の解析を困難にさせる．それにより，新しい機能を持つマルウェアが容易に拡散してしまう．

・ドロッパ

ドロッパは，マルウェアを内包した状態で流通し，ユーザの PC 上で実行されると，暗黙のうちにマルウェアをインストール（ドロップ）する機能を持ったマルウェアである．ドロッパの中には Microsoft Word 等の文書ファイルになりすまし，実行されると実際の文書を表示すると同時にマルウェアをインストールする巧妙なものも存在する．

2.1.4 「その他」のマルウェア

・ボット

ボットとは，ロボットの短縮語であり，指令者からの遠隔操作によって，多様な活動，目的，機能を実現するマルウェアである．ボットに感染した PC はボットネットと呼ばれるボット独自のネットワークを形成する．ボットネットは小規模なものでは数百，大規模なものでは数十万もの感染 PC 群によって成り立っている．指令者（攻撃者）は，IRC（Internet Relay Chat）サーバ（インターネットを通して文字による会話を提供するサーバ）経由でボットネットに命令し，その結果，多数のボットが命令に従って一斉に動作を行う．例としては，DDoS 攻撃がボットネットを用いて攻撃を行うものである．DDoS 攻撃は，ボットネット上の感染した複数の PC がスパムメールを大量に送信することで，通信路をあふれさせ，標的の機能を停止させてしまう攻撃である．

現在流通している Windows 等の OS は，そのコンピュータに行わせたい処理に応じて，ユーザがシステム内部の設定を自由に変更できるようになっている．そのため，利用者はネットワークを経由して多種多様のプログラムを容易に入手することができる．さらに，マルウェアは，技術と知識と時間があれば誰でも作成できる．これはつまり，マルウェアが無数に作成・存在できることを意味している．

このようなことから，近年においてコンピュータがマルウェアに侵される危険性が増しており，全世界を巻き込んだ社会問題となっている．

2.2 マルウェアの変遷・事件

マルウェアが登場した当初は，技術誇示的，愉快犯的動機による製作，散布が目的であった．しかし，PC，インターネットの普及に伴い，90年代後半以降は，電子メールによる大量複製・配布や自動感染型ウイルスの登場等，感染拡大による業務停止やウイルス駆除のため，多額の費用を支払わせることを目的としたネットワークを利用したものが新たに登場した．しかし，これらのマルウェアは依然として技術誇示的，愉快犯的動機が中心としたものであった．その後，マルウェア感染技術・手法が急速に進展し，ネットワーク感染型マルウェアの技術・手法を基盤に金銭的利益や犯罪行為，テロ目的に利用するために，特定の目的を持った機能を強化した，スパイウェア，ボット等のマルウェアが登場した．最近では，ネットワーク感染型のような自動感染ではなく，特定の目標を狙う標的型のマルウェアが話題となっている．このように，多様化・高度化・悪質化する不正なプログラムを統一的に表現する新たな用語が，学術的に，また社会的にも必要とされた結果，マルウェアという言葉が世界規模で認知され定着するようになった[5].

マルウェアに関する事件で最近話題になったものを以下にまとめた[6][7][8].

- 2011 年 7 月には，衆議院議員へのサイバー攻撃の事件が発生した．衆議院議員 1 人の公務用のパソコンに届いたメールの添付ファイルが開かれたことで，衆議院のサーバがトロイの木馬型のコンピューターウイルスに感染した．被害状況としては，攻撃者に議員や秘書ら約 2,700 人のメールや文書，ネットワーク利用者の ID とパスワードが盗み出された可能性があると言われている．

- 2011 年 9 月には，三菱重工業等日本の防衛産業の企業に対して，標的型攻撃のマルウェア感染が発生した．この攻撃はまず，攻撃者が，細工を施した PDF ファイルが添付された電子メールを防衛産業の企業に送りつける．もし古いバージョンの Adobe Flash／Reader を利用している社内の PC で PDF ファイルを開いてしまうと，脆弱性が悪用され，社内の PC がマルウェアに感染し，攻撃者のサーバとの接続を確立されてしまう．その後，攻撃者がそのセキュリティホールを使って，攻撃者のサーバから命令を行うことで，攻撃ツールやマルウェアを送りこみ，感染システムを制御できる状態にしてしまうという流れであった．また，攻撃に利用されたマルウェアは，それぞれの標的に合わせて作成されており，攻撃者側の意図的な攻撃であったと言われている．
- 2012 年末には，マルウェアに感染した PC が犯行予告や脅迫の書き込みの踏み台に利用され，その持ち主が逮捕され，一部は起訴や家裁送致されるという事件が発生しました．これは，犯人がインターネットの電子掲示板を介し，他者の PC を遠隔操作し，これを踏み台として襲撃や殺人などの犯罪予告を行ったサイバー犯罪（遠隔操作ウイルス事件とも呼ばれる）である．遠隔操作には，WEB サイトの脆弱性や，犯人が自作した未知のマルウェア（トロイの木馬（iesys.exe））が使用された．
- 2013 年は，ネットバンキングの ID・パスワードを盗み取られ，不正に現金を引き出される被害が急増した．ネットバンキングの暗証番号を盗み出す方法で，7 億 6,000 万円以上の被害が出ている．（2013 年 10 月時点）．被害が多くなった理由は，ネットバンキングを狙う不正プログラム「オンライン銀行詐欺ツール（Banking Trojan）」の感染が国内で広がっているためである．グラフはトレンドマイクロによる感染数の調査で，赤色の 2013 年に大きく増えていることがわかる（図 2.1）．累計では 2012 年の 7,375 台から，2013 年は 10 月末の時点ですでに 1 万 8101 台と，約 2.5 倍に増加していると，トレンドマイクロによって報告されている．

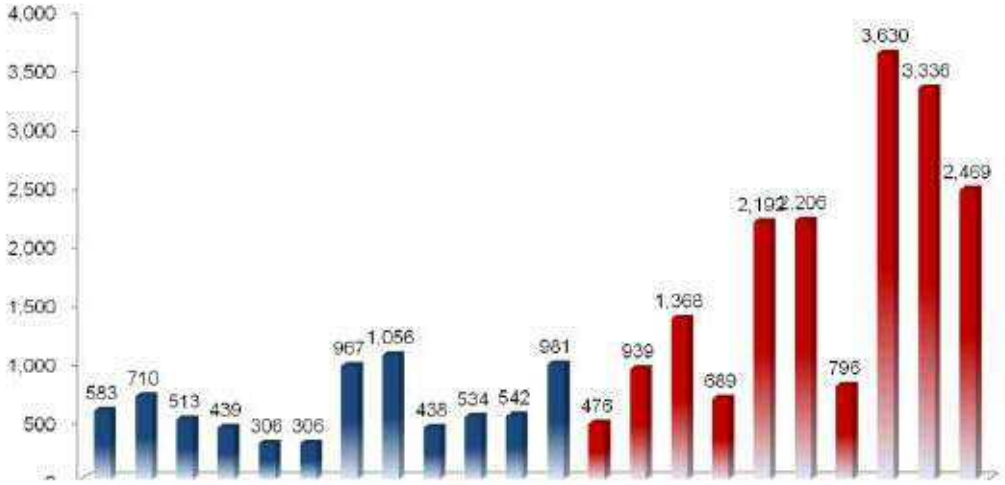

図 2.1 オンライン銀行詐欺ツール酷な感染台数推移（2013 年トレンドマイクロ）

2.3 マルウェア感染時の動作

マルウェア感染時の動作には，主に感染，攻撃，転送，接続の 4 つの活動に分類できる．

2.3.1 感染

マルウェアはプログラムやネットワークサービス等の脆弱性から感染していく．マルウェアの感染は，能動感染型と受動型感染の 2 つに分けられる[7].

(1) 能動感染型のマルウェアの感染とは，攻撃者が能動的に感染のための通信を開始する感染方式である．ネットワークサービスが動作しているサーバ等に対して，エクスプロイトコード（Exploit Code）と呼ばれる不正な動作を行うコードを送りこむ．脆弱性があるサーバであった場合，感染を受ける対象となる．

(2) 能動感染型のマルウェアの感染とは，サーバ等のノード管理者が自発的に悪意のあるプログ

ラムを実行し，ボットに感染する感染方式である．現在，多くの受動感染型のマルウェアはネットワーク経由あるいはメール経由で感染を行う．ネットワーク経由においては，ネットワーク上にある Web ページに Web ブラウザの脆弱性を利用したコードが含まれており，攻撃者がマルウェアを仕掛けた Web ページにアクセスした人に，不正な動作を行うコードを送りこむ手法である．また，メール経由においては，メールにマルウェアそのものあるいはマルウェアをダウンロードさせるためのソフトウェアを添付し，利用者に対して自発的にソフトウェアを実行させることによって感染させる手法である．

また，上記の感染は既知の攻撃と未知の攻撃に分類される．

既知の攻撃は，セキュリティ対策の製品開発や研究等を行っている企業（セキュリティベンダ）に認知されている攻撃を指す．セキュリティベンダは，攻撃の際に用いられる特徴や通信の特徴をシグネチャとして定義し，このシグネチャを用いて感染検知を行う．インターネットユーザは，セキュリティベンダが制作しているウイルス対策ソフト等の対策技術を正しく用いることで，十分な対策ができる．

未知の攻撃は，セキュリティベンダに認知されていない攻撃を指す．これらの攻撃は解析することで既知の攻撃となる．しかし，未知の攻撃はソフトウェアにセキュリティ上の脆弱性(セキュリティホール)が発見されたときに，問題の存在自体が広く公表される前にその脆弱性を悪用して行われる攻撃である．そのため，問題が発見してから修正プログラムが提供されるまでは攻撃を防ぐことができない．また，セキュリティベンダが問題を解析するために，数時間から数日程度かかってしまう．このように，解析から修正プログラムの提供までの時間差の問題から，セキュリティベンダによって解析が完了しても即座に攻撃を検知することは困難である．

2.3.2 攻撃

マルウェアの攻撃には，迷惑メールの送信，ディスクやファイル破壊，標的の脆弱性を狙った攻撃を行うもの等がある[6]．

(1) ポートスキャンを用いた感染被害の拡大

ポートスキャンとは，TCP/UDP のポートがオープンしているかどうか調査する行為のことを言い，どういうサーバが稼働しているか，外部からアクセス可能か，といった情報を知ることができる．ポートスキャンを用いることで，脆弱性のあるネットワーク上のホストを見つけ出すことができ，そのホストに対して，攻撃者がマルウェアや攻撃コードを含むコードを送りこむことで感染被害を拡大させる攻撃である．

(2) 標的型メール攻撃

特定の組織やコミュニティに狙いを定めて，文面，添付ファイル，使用ウイルス等を相手に合わせてカスタマイズすることで，感染確率，攻撃成功率を上げる手法．また，標的型メールは怪しさを感じさせない普通のメールであり，セキュリティベンダに対しての報告件数（配布件数）も少なく，感染後も目立つような活動を行わない巧妙な攻撃である．

(3) ドライブ・バイ・ダウンロード攻撃

Web サイト経由の攻撃手法の一つである．Web サイトを閲覧した際に，PC 利用者の意図に関わらず，ウイルス等の不正プログラムを PC にダウンロードさせる攻撃．不正プログラムをダウンロードさせられたことを利用者が気づかないため，被害が拡大しやすい傾向がある．また，ドライブ・バイ・ダウンロード攻撃では，主に利用者の PC の OS やアプリケーション等の脆弱性が悪用される．

(4) スパムメールの送信

スパムメール（迷惑メール）の送信とは，感染した PC から無差別にインターネットメールを送信する活動である．スパムメールは主に営利目的あるいは悪意を持つ Web サイトへの誘導のために送信される．営利目的のものでは，スパムメール送信者自らの利益となるような Web ページへの URL を記載したメールを送信する．悪意を持つ Web サイトへの誘導をするものでは，Web ブラウザや Web コンテンツが利用するアプリケーションの脆弱性を突く，攻撃コードを含んだ Web ページの URL を含むメールを無差別なユーザに対して送信する．

2.3.3 転送

攻撃者は，攻撃が成功したPC をボットとし，マルウェアの転送を開始させる[7]．なぜなら，

攻撃者は2.1.3 節で紹介したダウンローダ等のマルウェアを, 新たにPC に送りこむ必要があるからである．その理由として，攻撃者は一般的に文字列処理等，メモリ管理の脆弱性を突く攻撃をする．しかし，その攻撃手法ではソフトウェアの送信容量が制限されている．そのため，マルウェアを保管しているサーバから感染PC がダウンローダを利用することで，攻撃者はソフトウェアの送信容量の制限を解除する．

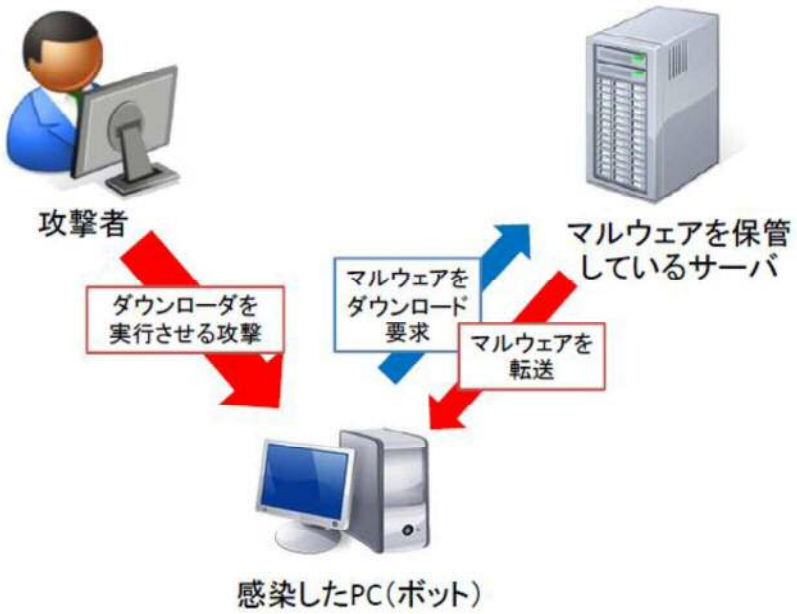

図 2.2　ダウンローダを用いたマルウェア転送の例

2.3.4　接続

感染したホストは特定サイトへの接続確認やインターネットの接続ポートを確認することで，活動までの準備を行っている[7].

[1] インターネット接続の確認
　感染した PC がインターネットに接続しているか確認する．マルウェアはコンピュータの外部 IP アドレスを取得し, そのコンピュータがインターネットへ直に接続しているかを確認する．すなわち，マルウェアは，攻撃コードをインターネット上に発信し，感染したコンピュータにインターネット接続があるかどうかを, 外部 IP アドレスをチェックすることで確認する．この確認は新たなマルウェアのダウンロードや C&C サーバへの接続のために行われている．

[2] ボットネット，C&C サーバに接続して指令を待つ
　ボットに感染した PC が行う接続である．ボットに感染した PC がインターネット接続の確認が完了した場合，指令を行う C&C サーバに接続する．以後は攻撃者がボットネットを管理しているサーバからの命令に従い活動する．

第 3 章　先行研究

現在行われているマルウェア対策研究として以下のものが挙げられる．

3.1　動的・静的解析

静的・動的解析は感染後の挙動の解析やコードの解析を行う研究である．この研究は「動的解析」と「静的解析」に分けられる．

3.1.1　動的解析

マルウェアのホストやネットワーク上での振る舞いを知る方法で，他のシステムに影響を与えない隔離された環境でマルウェアを実行し，その挙動に着目する解析手法である[7]．この手法はプログラムの実行された経路について挙動を容易に把握できるが，プログラムの仕様や，マルウェアが実行されていない処理を解析することが困難である．

3.1.2　静的解析

プログラムの構造や仕様に着目する解析手法である[7]．機械語を逆アセンブル（機械語で書かれたオブジェクトコードを，アセンブル言語で記述されたソースコードに変換すること）し，1 命令ずつ解析することで，プログラムの仕様を完全に把握することが可能である．

しかし，完全な解析を行うためには相応の知識・経験および時間が必要となる．上記の動的解析はプログラムの概要を把握するためには有用であるが，正確な解析を行う事ができない場合がある．そのため，通常は動的解析を行った上で，静的解析による詳細な解析を実施することが多い．

3.2　ハニーポット観測

静的・動的解析等の研究でマルウェアの解析を行うために，マルウェアを収集する方法としてハニーポッド観測が挙げられる．ハニーポットとは，クラッカーからの侵入方法やウイルスの振る舞い等を調査・研究するためにインターネット上に設置された，故意に侵入者を容易にできるように設定された囮システムである[9]．また，ハニーポットを設置したユーザは，囮として PC を設置したため，その PC を使用しない．そのため，ハニーポット内のファイルシステムは変更されない．もし変更が加えられた場合は何らかの攻撃を受けシステムを侵害されたと考えられる．ハニーポットはネットワーク越しには単なる脆弱なサーバやネットワーク機器にしか見えないが，侵入しても外部に攻撃できないような設定がされている．また，侵入者やウイルス等が介入できない方法で詳細な情報を記録している．

ハニーポットにおびき寄せられた侵入者の行動記録を調査することによって，コンピュータへの侵入や攻撃の手法を研究し，早急にウイルスを捕獲して分析することが可能である．

しかし，ハニーポット観測で収集されるデータは莫大な量であり，そのデータを基に攻撃の手法や不正な行為の証拠を特定するには，相応の知識・経験および時間が必要となる．

3.3　感染検知

マルウェアの解析の結果，得られた知見を用いて，感染検知を行う．本研究で対象とするマルウェア感染検知に関連する従来の研究を紹介する．

3.3.1　シグネチャ型検知

マルウェアコード内の特徴的な部分を「パターン」として取り出してデータベース化しておき，それを検索対象のファイル内容と照合する方法である．同じコードを持っていた場合，そのファイルはマルウェアであると特定する[10].

この方式の問題点は，検索ソフトのパターンファイルに登録されていないものはマルウェアとして検出できないことである．そのため，常に新しいパターンファイルを更新していく必要がある．また，解析されていない新種のマルウェアに対しても効果がない．

3.3.2　ルールベース型検知=

マルウェアの活動を分析してルール化しておき，プログラムの動作を監視し，プログラムの動作がルールと合致していればマルウェアとして判定する方法である．「ルール」にしたがってマルウェア検出を行うので「ルールベース」と呼ばれている[10].

ルールベース型検知は，解析済みのマルウェアばかりではなく，解析が済んでいないマルウェアに対しても有効である．また，マルウェアが感染発病活動を行う前に，その活動を未然に防ぐことができる．しかし，問題としては，マルウェアに似た活動をする正規のアプリケーションがあり得るため，本当にマルウェアかどうかの確認が難しく，正常な通信を誤検知してしまう可能性がある．

3.3.3　イベント観測型検知

マルウェア感染後の動作として現れる，DDoS 攻撃やスパムメール発信等の感染後の挙動に着目したホストを検知する方法である[11].

この方式の問題点としては，マルウェア自身が本来持っている機能が動作し始めてからの検知になり，情報が膨大になるため，解析が困難になってしまう問題がある．

3.3.4　ヒューリスティック検知

実行ファイルの挙動等を解析し，ライブラリファイルの書き換え等，一般的にはあまり見られない特異な挙動から未知のマルウェアを検知する方法である[2].

この方式の問題点としては，未知のウイルスを確実に発見できるわけではなく，また，ウイルスではないものをウイルスと判定し，誤検知をしてしまう問題点がある．

3.1～3.3 節の中で，本研究ではマルウェア感染の被害拡大を最小限に抑えるために実用的であると思われる感染検知に着目した．また，感染検知の中でも，未知マルウェア検知に最も実用的であると思われるトラヒックデータ観測検知について着目した．

3.4　トラヒックデータ観測による感染検知

トラヒックデータ観測とは，ボットネットにおける C&C (Command & Control)サーバと感染した PC 間の通信や感染後の DNS（Domain Name Server）クエリの異常等を検知する感染検知手法である[11].

3.4.1　トラヒックデータ

トラヒックデータとは，通信に利用されているデータであり，パケットの束が集まることによって構成されている．例としては，データ量，送受信者の位置，経路，日付，時刻，電子メールアドレス，ウェブサイトアドレス等の情報がトラヒックデータに含まれている．

3.4.2　トラヒックデータに着目する理由

トラヒックデータに着目する理由としては，正常時と感染時のトラヒックデータの特徴に違いが見られたためである．この違いに対し，今回用いるパターン認識の技術を適用することができると考えられる．

また，この検知手法は 3.3 節で説明した検知手法と比べて，トラヒックデータの情報から得られる特徴量の時間的な変化を見ることによって，検知性能の向上の可能性がある．また，過去のマルウェアに感染したデータを学習することで，既知の検体の検知だけでなく，未知の検体の検知を行うことができる可能性がある．

3.4.3　トラヒックデータの構成

トラヒックデータはパケットという単位に分割されて送受信が行われている．通信パケットは主に，ヘッダ部とペイロード部に分けることができる．

3.4.3.1　ヘッダ

ヘッダとは，通信データの先頭に付加される制御情報である．ヘッダはネットワークで使われ

る通信プロトコルごとに定まっており，それぞれのヘッダの構成はプロトコルが提供する機能を実現できるように工夫して作られている．例としてはEthernet，IP（Internet Protocol），TCP（Transmission Control Protocol）等のプロトコルでは，パケットの送信元や宛先のアドレス等の情報をヘッダに格納している．

3.4.3.2　ペイロード

ペイロードとは，通信パケットのうちヘッダ部分（行先等の付加情報）を除いた，本来送信者が特定の受信者に転送したいデータ本体のことである．

ネットワーク上を流れるデータは基本的に「ヘッダ＋ペイロード」という構成になっており，ヘッダがパケット自体を転送するために必要な情報を含んでいるのに対し，ペイロードは転送を依頼したアプリケーションソフトが実際に転送したい情報を含んでいる．小包に例えると，ヘッダは荷札で，ペイロードは梱包された荷物，といった関係になっている．

3.4.4　ヘッダ部分からの検知

トラヒックデータのヘッダ部分を用いた感染検知・異常検知の研究事例を紹介する．

藤原[12][13]らは，ポートスキャンを行っているPCのトラヒックデータの挙動に着目する手法を提案した．これは，トラヒックデータのパケット数，パケットサイズを特徴量として，正常時及び感染時トラヒックデータの1秒ごとに送受信されたパケットサイズを合計したグラフを生成し，正常時及び感染時通信の時間推移に対するパケットサイズをあらかじめ学習した識別器（識別辞書）で比較（パターン認識）を行うことで，入力されたトラヒックデータが正常か異常かを判定した．結果としては，ポートスキャンの特徴として，一定サイズの連続するパケットの送信が特徴であることを確認でき，時間的推移に着目したパケットサイズを用いた感染検知は有効であると示している．

宮本[14]らは，攻撃者による不正侵入行為を行っているトラヒックデータに着目する手法を提案した．これは，パケットのヘッダ情報（TCPフラグ別パケット数，TCPプロトコル別パケット数，パケットサイズ別パケット数）を種別ごとにカウントし，トラヒックデータを多次元ベクトルデータとして数値化し，作成されたベクトルデータに対してSVM（Support Vector Machine）を用いたクラスタリング手法を適用することにより，未知の侵入行為を行う新種・亜種のマルウェアを検出できる異常検知手法である．結果としては，低レイヤの異常に対して適用した結果，異常検出手法として有効であると示している．

従来，ネットワークの異常検知という分野がある．通常とは異なる振る舞いを検出することで異常を検知している．マルウェアの感染時を「正常ではない状態」と捉えることも出来るため，異常検知手法も参考になると考えられる．

3.4.5　ペイロード部分からの検知

トラヒックデータのペイロード部分を用いた感染検知・異常検知の研究事例を紹介する．

桑原ら[15]はボットの攻撃通信データのペイロード部分から，マルウェアの挙動と対応した特徴的な文字列に着目する手法を提案した．トラヒックデータから特徴的な文字列の出現回数や，特定の文字列が出現した後に現れる挙動を特徴量として，それらの特徴量を分岐とした決定木を生成し，その決定木を用いてマルウェアの種類の判定を行った．結果としては，マルウェアのダウンロードにおいて文字列だけでは通常の通信を含めた場合の感染判定は困難であり，他の検知手法との組み合わせにより，検知率を向上させられることを示している．

Wei Lu[16]らは，ボットの制御通信を行っているトラヒックデータに着目する手法を提案した．これは，宮本ら[11]と同様に，正常時通信とは異なる振る舞いを検出することでボットが行っている通信（異常）を検知している．正常時通信と感染時通信パケットのペイロード部に着目し，ペイロードの内のASCII文字コードの発生頻度，平均値，標準偏差を特徴量として，クラスタリング手法を適用することでボットの挙動を形式化し，新種・亜種のマルウェアを検出できる異常検知手法である．結果として，この検知手法は高い検知率を取得することができ，有効であることを示している．

山田[17]らは，侵入検知システムの未知攻撃の課題に対する解決策を示している．侵入検知システムには，3.3節で説明したシグネチャ検知と正常な通信と異なる状態をアノマリ（Anomaly）として検知するアノマリ検知に分類される．侵入検知システムは，シグネチャ検知において未知

攻撃を検知することが全くできず, アノマリ検知において誤検知を多く発生してしまう. しかし, 事前にシグネチャ検知の結果を用意し, 機械学習を行ったアノマリ検知は誤検知が少なくなる. そこで, 機械学習によるアノマリ検知において, シグネチャを効率的に生成することによって, 未知攻撃の検知精度を向上させる提案をした. 方法としては, HTTPリクエストに含まれるHTTPリクエストラインのサイズ, field-nameに対するfield-valueのサイズ, HTTPリクエストの総サイズを特徴量として, 正常と異常を正確に分類できる決定木を機械学習として採用し, 未知攻撃の検知を行った. 結果としては, 提案方法により, 未知攻撃に対する検知手法として有効であることを示している.

本研究では, ヘッダ部分ではなくペイロード部分に着目した. ペイロード部分は, ヘッダ部分よりも情報量が多く, 正常時の通信と感染時の通信を区別化する情報が多く含まれていると考えられる. そのため, 情報量が限られているヘッダ情報を用いた感染検知よりも, 情報量が多いペイロード情報を用いた感染検知の方が, 多くの種類のマルウェアを検知できる可能性がある. また, ヘッダ情報を用いた感染検知では検知できないマルウェアを, ペイロード情報を用いた感染検知と組み合わせることで, 検知率が向上できることが考えられる.

第 4 章　トラヒックデータの特徴量の分析

4.1　既存研究の特徴量

本章では，既存のマルウェア感染検知やネットワーク異常検知に関する研究で用いられているペイロードの特徴量を整理する．

文献[12]，[14]，[15]等では，特徴量として文字列の出現頻度が使われている．桑原ら[15]は，ボットの攻撃通信データのペイロード情報から，マルウェアの挙動毎に対応した特徴的な文字列（exe, NICK 等）があることを確認し，それらを特徴量として用いている

また，Wei ら[16]は，ボットが行う遠隔制御用通信のトラヒックデータに着目する手法を提案し，正常時通信とボットが行っている感染（異常）時通信パケットのペイロード情報に着目しペイロード内の ASCII 文字コードの出現頻度（バイト数）を特徴量としている．

また，山田ら[17]は，侵入検知システムにおける未知攻撃の課題に対する解決策として，決定木を用いたアノマリ検知を示し，HTTP リクエスト長，HTTP リクエストの総サイズを特徴量としている．

上記より，マルウェア感染検知用の特徴量として ASCII 文字コードの出現頻度，文字列の出現頻度，HTTP リクエスト長がよく用いられていることがわかる．しかし，いずれの研究においても，個々の特徴量の有効性については評価されていないため，個々の特徴量の有効性を評価する必要がある．

4.2　マルウェアの種類毎の特徴量

トレンドマイクロ社のセキュリティデータベース[18]等に示されているように，マルウェアの種類毎で特有の挙動がある．例として，ワームはソフトウェアに存在する脆弱性を利用し，ネットワーク上で感染活動を行う．トロイの木馬は特定のサイトにアクセスし，不正なファイルのダウンロードを要求し，感染した PC がさらなる脅威にさらされるなどの挙動がある．同種類のマルウェアの亜種で共通した挙動の傾向があることが確認できている[18]．このため，マルウェアの種類毎に有効な特徴量を求める事ができる可能性がある．

第 5 章　特徴量評価実験

5.1　実験概要

マルウェアの種類毎に，ペイロードの個々の特徴による正常と感染の区別に対する有効性を明らかにするために，ペイロードに含まれる特徴量の観点から分析する．これらの個々の特徴量の有効性を明らかにする事で，マルウェアの種類毎に，検知に最適な特徴量の組み合わせを明らかにする．本論文では，研究の第一段階として，個々の特徴量のマルウェアの種類に対する有効性を明らかにする（図 5.1）．

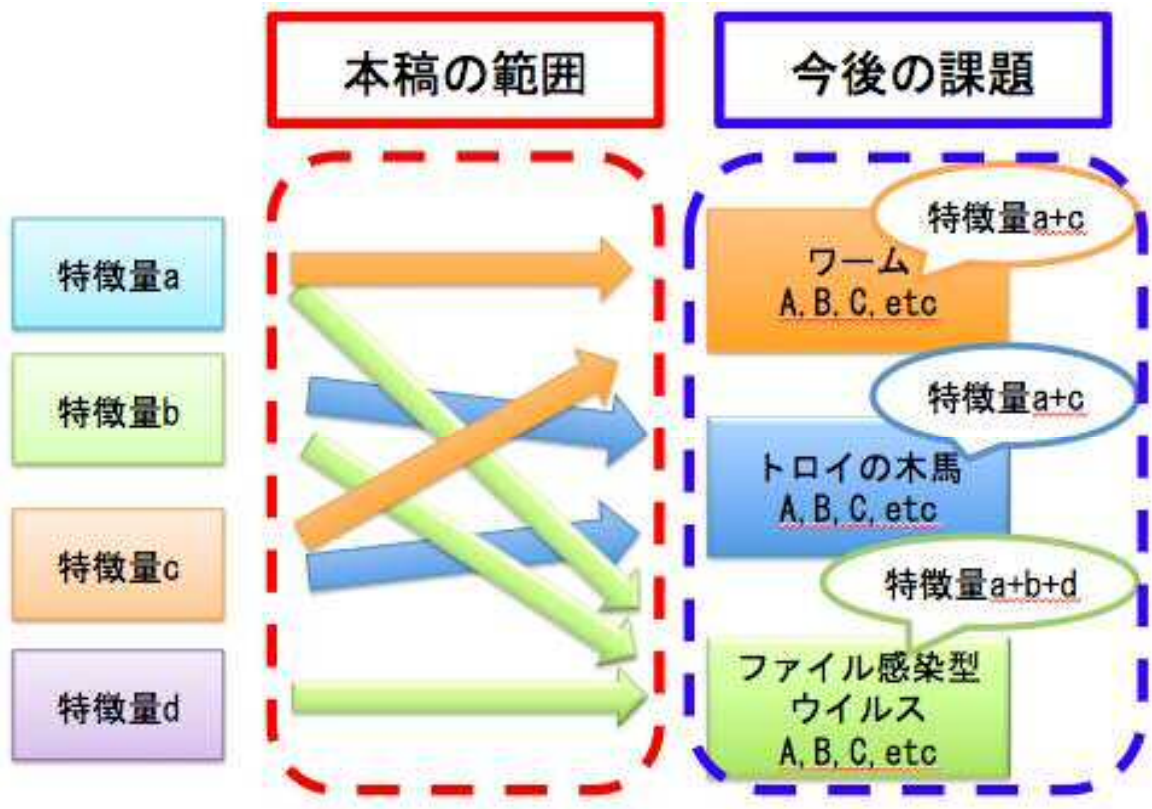

図 5.1 本稿の範囲と今後の課題の範囲

5.1.1　ベクトル量子化

本研究で使用したベクトル量子化について以下に示す．ベクトル量子化（Vector Quantization）とは，K 個の信号をまとめてひとつの K 次元ベクトル，すなわち K 次元信号空間内の一点とし，あらかじめ定められたいくつかの代表点(量子化代表ベクトル)で近似するものである[19]．
まず入力ベクトルを K 次元ベクトルとする．このとき，入力ベクトルが存在する信号空間は K 次元となる．この K 次元空間を $x = (x_1, x_2, \cdots, x_K)^T$ と書くことにする．ここで K 次元信号空間 R^K を互いに重なり合わない N 個の領域 $P_1, P_2, \cdots, P_N$ を分割し，各領域 P_i 内に量子化代表ベクトル $y = (y_{i1}, y_{i2}, \cdots, y_{iK})^T$ を 1 つ定めておく．ここで，領域 $P_1, P_2, \cdots, P_N$ の集合を P と書き，分割と呼ぶ．こういった記法を用いると，「K 次元 N レベルのベクトル量子化」は K 次元信号空間 R^K からコードブック C への写像 $Q(\cdot)$ として，次のように記述される．

ベクトル量子化の動作：
もし入力ベクトル x が領域 P_i 内に所属しているならば，そのときのベクトル量子化の動作は次式の写像 $Q(\cdot)$として記述される．

$$Q(\cdot) = y_i \tag{5.1}$$

式(5.1)のベクトル量子化によるサンプル辺りの平均ひずみ D は，入力ベクトル x の確率密度関数 $p(x)$ が既知の場合には次式で与えられる．

$$D = \frac{1}{K}\int_{R^k} d\big(Q(x)\big)\cdot p(x)dx$$
$$= \frac{1}{K}\sum_{i=1}^{N}\int_{R^k} d(x,y_i)\cdot p(x)dx \tag{5.2}$$

次元数 K とレベル数 n とが指定された時，式(5.2)の平均ひずみ D を最小とするベクトル量子化器のことを「最適なベクトル量子化器」を呼ぶ．式(5.2)において平均ひずみ D の大きさは領域 P_i や量子化代表ベクトル y_i をどのように設定するかに左右される．従って，ベクトル量子化器を最適化するためには，領域や量子化代表ベクトルを式(5.2)の D を最小とするように定めなければならない．このための必要条件は，以下に示すような 2 つの最適化条件に帰着する．

1.　代表点条件
K 次元信号空間 R^K の N 個の領域 $P_1, P_2, \cdots, P_N$への分割 P が与えられたとき，式(5.2)の平均ひずみを最小とする N 個の量子化代表ベクトル $y_1, y_2, \cdots, y_N$はそれぞれ対応した領域 $P_1, P_2, \cdots, P_N$ の重心で与えられる．入力ベクトル x の確率密度関数$p(x)$ が既知の場合は，量子化代表ベクトル y_i が次式で計算されることを意味する．

$$D = \frac{\int_{P_i} x\cdot p(x)dx}{\int_{P_i} p(x)dx} \tag{5.3}$$

2.　分割条件
N 個の量子化代表ベクトル $y_1, y_2, \cdots, y_N$ から成るコードブック C が与えられた時，式(5.3)の平均ひずみ D を最小とする N 個の領域 $P_1, P_2, \cdots, P_N$への K 次元信号空間 R^K の分割 P は，次のようにして与えられる．すなわち，量子化代表ベクトル $y_1, y_2, \cdots, y_N$に対応した領域 P_i は，N 個の量子化代表ベクトルの中で y_iとのひずみが最小となる入力ベクトル x，すなわち y_i 以外のすべての量子化代表ベクトル $y_i(j \neq i)$に対して式の不等式

$$d(x,y_i) \leq d\big(x,y_j\big) \tag{5.4}$$

を満足する入力ベクトル x の集合として与えられる．
また，この条件によって式(5.1)の写像 $Q(\cdot)$として定義された量子化動作を次のように書き換える事ができる．

最適なベクトル量子化器の量子化動作：
入力ベクトル x が y_i以外の全ての量子化代表ベクトル $y_i(j \neq i)$に対して式(5.4)の不等式を満足するならば，その時最適なベクトル量子化器の量子化動作は，入力ベクトル x から量子化ベクトル y_i への写像として，$Q(x) = y_i$と記述される．また，本研究ではひずみ測定に，入力ベクトルと量子化代表ベクトルの間のユークリッド距離の 2 乗として定義された 2 乗ひずみ測定を用いている．

5.1.2　クラスタリングアルゴリズム

本研究では，クラスタリングアルゴリズムとして，LBG+splitting アルゴリズムを用いている．

■　LBG アルゴリズム

LBG アルゴリズムは，適当なコードブックから出発し，学習系列に分割条件として代表点条件を繰り返し適用し，良好なコードブックに収束させる設計アルゴリズムである[20]．その処理の流れを図 5.2 に示し，手順を以下に示す．

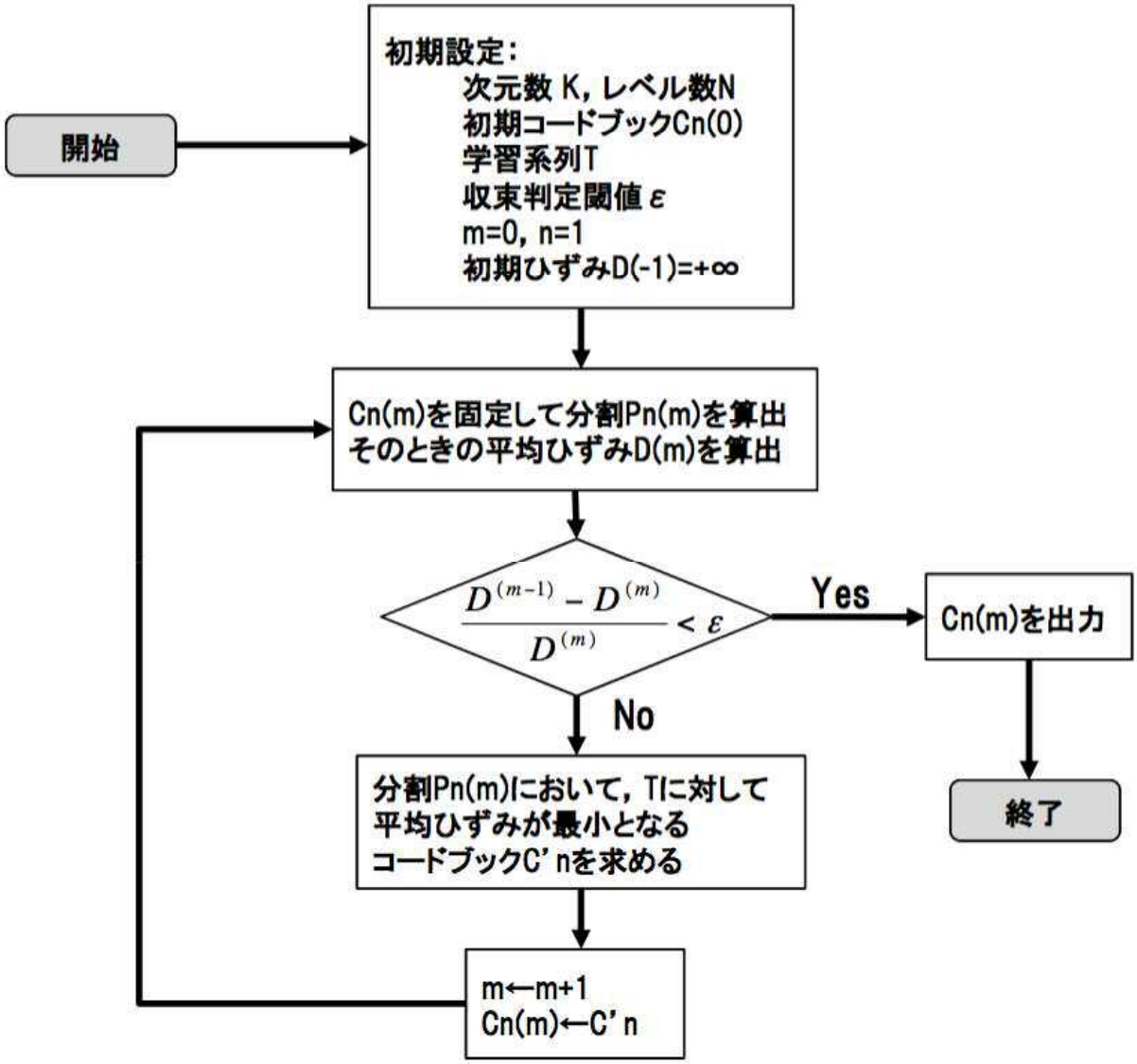

図 5.2　LBG アルゴリズムにおける処理のフローチャート

1. 次元数 K，レベル数 N，N 個の初期量子化代表ベクトル$y_1^{(o)}, y_2^{(o)}, \cdots, y_N^{(o)}$からなる初期コードブック$C_N^{(o)}$，L 個の K 次元学習ベクトル$x_1^{(o)}, x_2^{(o)}, \cdots, x_N^{(o)}$からなる学習系列 T，収束判定用閾値 ε が与えられているとする．また$m = 0$，初期ひずみ値 $D^{(-1)} = \infty$と設定する．
2. 量子化代表ベクトル$y_1^{(m)}, y_2^{(m)}, \cdots, y_N^{(m)}$からなる初期コードブック$C_N^{(m)}$の下で，平均ひずみを最小とする N 個の量子化代表ベクトルの中で，$y_i^{(m)}$とのひずみが最小となる学習ベクトルの集合で与えられる．こうして，L 個の学習ベクトルが N 個の領域に分割される．また，各領域に所属する学習ベクトルを，その領域内の量子化代表ベクトルで置き換えたときに生じる平均ひずみ$D^{(m)}$を算出する．
3. もし，$\left(D^{(m-1)} - D^{(m)}\right)/D^{(m)} < \varepsilon$ ならば，処理を停止して，$C_N^{(m)}$ を最終的に設定された N レベルのコードブックとして出力する．さもなければ次の手順に進む．
4. N 個の領域$P_1^{(m)}, P_2^{(m)}, \cdots, P_N^{(m)}$への学習系列 T の分割$P_N^{(m)}$の下で，学習系列 T に対して平均ひずみを最小とする N 個の最小とする N 個の量子化代表ベクトル$y_1^{(m)}, y_2^{(m)}, \cdots, y_N^{(m)}$から成るコードブック$C_N'$を代表点条件に適用して定める．領域$P_i^{(m)}$に所属する学習ベクトルの平均ベクトルとして与えられる重心を，量子化代表ベクトルy_i'とする．さらに，$m \leftarrow m + 1$ とし，C_N'をコードブック$C_N^{(m)}$として，手順 2.へ戻る．

■ Splitting アルゴリズム

LBG アルゴリズムにより設計されたコードブックの良否は，初期コードブック$C_N^{(0)}$と学習系

列 T の選定法に強く依存する．初期コードブック$C_N^{(0)}$は想定される入力ベクトルの分布範囲を被覆していることが望ましい．この条件をある程度満足する初期コードブックの生成法として splitting アルゴリズムが知られている．このアルゴリズムは，N レベルのコードブックC_Nの量子化代表ベクトル$y_1, y_2, \cdots, y_N$を式(5.5)のように微小なベクトルδ を用いて接近した 2 つのベクトルに分割することによって，量子化代表ベクトル$y_1, y_2, \cdots, y_{2N}$からなる 2N レベルの初期コードブック$C_{2N}^{(0)}$を生成するものである．この splitting アルゴリズムを LBG アルゴリズムと組み合わせることによって 1 レベルのコードブックから出発して順次 2,4,8,・・・レベルのコードブックを設計することができる．

$$y_i = y_i - \delta,\ y_{i+N} = y_i + \delta \tag{5.5}$$

5.1.3 タイムスロット

本研究では，トラヒックデータからの特徴抽出にタイムスロットを用いた．タイムスロットとは，トラヒックデータを特定の秒数で区切った範囲のことを示す.

時間的変化を伴う通信をタイムスロットで分割し, タイムスロットの時間的変化を追うことで, 通信全体の時間的変化に着目した識別を行うことができる. 本実験では, タイムスロット幅は1 秒とし，タイムスロット毎に特徴量を求めた.

5.1.4 実験に用いるペイロードの特徴量

本研究では既存研究で多く用いられていた次に示す 261 種類の特徴量を評価対象とする.

- ASCII 文字コードの出現頻度 255 種類
- 特徴的な文字列の出現頻度：5 種類
  （GET，POST ，exe，whatismyip，checkip）
- HTTP リクエスト長

5.1.5 実験データ

本研究では，取得環境の違いを評価するために，正常時通信として，異なる 2 カ所のイントラネットに流れる通信を用いた．これにより，取得環境の違いによる影響を受けにくい特徴量を評価できる．それぞれのデータを，正常コードブック作成のための学習データとテストデータ用に分割している.

また，感染時通信に D3M2012，CCC2009, CCC2010, CCC2011 を用い，感染コードブック作成用の学習データとテストデータ用に分割した[21][22]. さらにこれらのデータをマルウェアの種類毎でマルウェアの検体数が均等になるように分割した．例えばワームの場合は，4 つのデータセットのうち，3 つのデータセットに 6 検体のワームがある．これらのワームをテストデータと学習データで 3 対 3 になるように分類した．それにより，学習した 3 つのデータから別の種類の 3 種類を検知できるかを評価した.

また，CCC2009， CCC2010, CCC2011 の攻撃通信データにはマルウェアに感染するまでの通信が含まれている．そこで本研究では，文献[23]と同じ方法を用いて，攻撃通信からマルウェアに感染した後の通信のみを切り出し評価した.

本研究ではマルウェアの種類をワーム，トロイの木馬，ファイル感染型ウイルスとした．ファイル感染型ウイルスとは，拡張子 exe 等の実行型ファイルに感染するウイルスである．また，マルウェア検体名は CCCDATAset において，攻撃元通信データのログファイルに記載されている名称を用いた．D3M2012 においては，マルウェア検体のハッシュ値を用い，G-data 社，トレンドマイクロ社，Kaspersky 社のそれぞれが命名しているマルウェア検体の名称を用いた．表 1 に今回実験で使用した各データセットのマルウェアをまとめた.

表 5.1：各データセットのマルウェア

種類	CCCDATAset2011	CCCDATAset2010	CCCDATAset2009	D3M2012
ワーム	WORM_DOWNAD.AD	WORM_DOWNAD.AD WORM_MAINBOT.AH WORM_MAINBOT.FY WORM_PALEVO.SMD WORM_PALEVO.BL	WORM_SWTYMLAI.CD	
トロイの木馬			TROJ_BUZUS.AGB	TROJ_GEN.R47C7C2 Trojan.Generic.KD.578000 Trojan.Generic.KD.410743 Trojan-Dropper.Win32.Dapato.aoxn
ファイル感染型		PE_VIRUT.AV PE_VIRUT.XV	PE_BOBAX.AK	

5.1.6　評価方法

本研究での感染時通信と正常時通信の識別方法について説明する．

5.1.6.1　ベクトル量子化によるコードブック作成

ベクトル量子化を用いて，感染時通信のみを用いて学習を行った感染コードブックと，正常時通信のみを用いて学習を行った正常コードブックを予め作成する．今回は，各特徴量を個別に評価することが目的であるため，特徴量毎に 1 次元コードブックを作成した．ベクトル量子化のアルゴリズムには，LBG+Splitting アルゴリズムを用い，レベル数は 2，4，8，16 の 4 種類とした．

5.1.6.2　コードブックを用いた計算方法

ベクトル量子化を用いて作成したコードブックを用い，テストデータと感染，正常コードブックとの距離を計算し，感染(正常)コードブックとの距離の方が小さければ感染(正常)と識別した．

第 6 章　特徴量評価実験の結果

ベクトル量子化を用いてトラヒックデータを正常と感染に分類し，これらが正しく分類されているかによって，特徴量の評価を行う．

特徴量は，検知率と誤検知率によって評価される．

検知率（True Positive Rate （以下 TPR）& True Negative Rate （以下 TNR））は感染（正常）時通信を正しく検知する確率である．また，1 から検知率を引いた値である見逃し率（False Negative）が用いられること場合もある．

誤検知率（False Positive）は，正常（感染）時通信を誤って感染（正常）として判定する確立である．

Positive（陽性）とNegative（陰性）とは，異常ありと異常なしに判定することを意味する．また，True（真）とFalse（偽）は，正しく判定することと誤って判定することを意味する．つまり，False Positiveの場合，異常ありと判定し，それが誤っていることになる．

また，検知率と誤検知率を式に表すと式(6.1)のようになる．

$$\begin{aligned}(\text{検知率}) &= 1-(\text{False Negative Rate}) \\ &= \frac{\text{True Positive(or True Negative)}}{(\text{True Positive})+(\text{False Negative})} \\ &= \frac{\text{感染（正常）と正しく識別できるサンプル数}}{\text{全感染（正常）サンプル数}}\end{aligned} \tag{6.1}$$

本研究では，トラヒックデータを正常と感染に分類し，正常時通信と感染時通信の特徴量が正しく「感染」と「正常」に分類されているかを，式(6.1)の検知率 TPR，TNR を用いて特徴量の評価を行う．

マルウェア感染検知の要件は，感染と正常を正しく識別できる特徴量を用いることである．このため，感染・正常のみを正しく識別できる特徴量を併用することも有効である．そこで，TPR（感染データを感染と正しく分類した割合），TNR（正常データを正常と正しく分類した割合）が共に高い特徴量，TPR のみが高い特徴量，TNR のみが高い特徴量の観点から感染検知に有効な特徴量について検討した．以下に評価実験の結果を示す．

6.1　TPR,TNR 共に高い特徴量

取得環境の影響を受けにくい特徴量を評価する観点から，2 種類の正常時通信を用い，量子化レベル数の影響を評価するために， 4 つの量子化レベル数における TPR，TNR の平均値を特徴量 261 個に対してそれぞれ求めた．その結果の中から，表 6.1 はイントラネット A を用いたときの平均 TPR・TNR の値が高い上位 15 個をマルウェアの種類毎に示している．さらに表 6.1 ではこれらの 45 個の特徴量についてイントラネット B の TPR・TNR を示す．表 6.1（網掛けがついている項目）より，ワームの ASCII 文字コード「i」とファイル感染型ウイルスの「HTTP リクエスト長」の TPR・TNR の値が，取得環境に依らず安定して高い値を示すことがわかった．これらの特徴量を用いたときの詳細を表 3，表 4 に示す．これらの特徴量は 2 種類の正常時通信のいずれの場合でも，量子化レベル数を変化させても，TPR が 95%以上かつ TNR が 80%以上で安定的に検知できることがわかった．

6.2　TPR のみ高い特徴量

表 6.1（太点線で囲んだ項目）から TPR のみ高い特徴量として，ワームでは HTTP リクエスト長, ASCII 文字コード「0」,「f」, トロイの木馬では HTTP リクエスト長, ASCII 文字コード「NL*」,「CR」,「0」,「5」,「A」,「C」,「M」,「d」,「e」,「r」,「t」,「x」, ファイル感染型ウイルスでは ASCII 文字コード「S」,「e」,「i」,「o」,「p」が確認できた．これらは 2 種類の正常時通信のいずれの場合でも，TPR が 95%以上で安定的に検知できることがわかった．

6.3　TNR のみ高い特徴量

表 2（細点線で囲んだ項目）から TNR のみ高い特徴量として，ワームでは ASCII 文字コード

「J」が確認できた．これらは 2 種類の正常時通信のいずれの場合でも，TNR が 80%以上で安定的に検知できることがわかった．

表 6.1：イントラネット毎の平均 TPR・TNR

マルウェアの種類	特徴量	イントラネットAの平均TPR	イントラネットBの平均TPR	特徴量	イントラネットAの平均TNR	イントラネットBの平均TNR
ワーム	HTTPリクエスト長	100%	99%	ASCII文字コード「>」	89%	79%
	ASCII文字コード「i」	99%	98%	ASCII文字コード「!」	88%	78%
	ASCII文字コード「f」	98%	97%	ASCII文字コード「ETB」	88%	70%
	ASCII文字コード「C」	98%	92%	ASCII文字コード「J」	85%	83%
	ASCII文字コード「E」	97%	94%	ASCII文字コード「H」	84%	74%
	ASCII文字コード「e」	97%	91%	ASCII文字コード「#」	84%	78%
	ASCII文字コード「a」	96%	91%	ASCII文字コード「,」	84%	76%
	ASCII文字コード「0」	95%	97%	ASCII文字コード「B」	84%	74%
	ASCII文字コード「r」	95%	91%	ASCII文字コード「E」	84%	75%
	ASCII文字コード「CR」	95%	91%	ASCII文字コード「P」	84%	75%
	ASCII文字コード「/」	95%	92%	ASCII文字コード「R」	84%	74%
	ASCII文字コード「:」	95%	94%	ASCII文字コード「S」	84%	73%
	ASCII文字コード「c」	95%	93%	ASCII文字コード「j」	83%	83%
	ASCII文字コード「s」	95%	94%	ASCII文字コード「M」	83%	75%
	ASCII文字コード「m」	95%	91%	ASCII文字コード「N」	83%	77%
トロイの木馬	HTTPリクエスト長	100%	100%	ASCII文字コード「<」	85%	72%
	ASCII文字コード「NL*」	100%	100%	ASCII文字コード「US」	85%	76%
	ASCII文字コード「CR」	100%	100%	ASCII文字コード「>」	85%	41%
	ASCII文字コード「0」	100%	100%	ASCII文字コード「¥」	83%	73%
	ASCII文字コード「5」	100%	100%	ASCII文字コード「VT」	82%	56%
	ASCII文字コード「C」	100%	100%	ASCII文字コード「]」	82%	68%
	ASCII文字コード「d」	100%	100%	ASCII文字コード「HT」	81%	65%
	ASCII文字コード「e」	100%	100%	ASCII文字コード「SOH」	80%	56%
	ASCII文字コード「t」	100%	100%	ASCII文字コード「^」	80%	69%
	ASCII文字コード「x」	100%	100%	ASCII文字コード「'」	80%	67%
	ASCII文字コード「A」	100%	100%	ASCII文字コード「FS」	80%	63%
	ASCII文字コード「o」	100%	100%	ASCII文字コード「ESC」	79%	62%
	ASCII文字コード「r」	100%	100%	ASCII文字コード「ACK」	79%	63%
	ASCII文字コード「1」	100%	100%	ASCII文字コード「$」	79%	65%
	ASCII文字コード「2」	100%	100%	ASCII文字コード「EOT」	79%	60%
ファイル感染型ウイルス	HTTPリクエスト長	100%	100%	ASCII文字コード「DC2」	92%	68%
	ASCII文字コード「p」	99%	99%	ASCII文字コード「DC3」	90%	68%
	ASCII文字コード「B」	99%	75%	ASCII文字コード「ETB」	89%	62%
	ASCII文字コード「S」	98%	98%	ASCII文字コード「#」	89%	67%
	ASCII文字コード「e」	98%	98%	ASCII文字コード「S」	89%	69%
	ASCII文字コード「T」	98%	94%	ASCII文字コード「Y」	88%	76%
	ASCII文字コード「i」	98%	98%	ASCII文字コード「'」	87%	79%
	ASCII文字コード「o」	97%	98%	ASCII文字コード「M」	87%	79%
	ASCII文字コード「H」	96%	94%	ASCII文字コード「R」	86%	79%
	ASCII文字コード「X」	95%	74%	ASCII文字コード「US」	86%	69%
	ASCII文字コード「W」	95%	93%	ASCII文字コード「)」	85%	75%
	ASCII文字コード「r」	95%	94%	HTTPリクエスト長	82%	84%
	ASCII文字コード「G」	95%	78%	ASCII文字コード「(」	82%	78%
	ASCII文字コード「N」	95%	80%	ASCII文字コード「f」	82%	73%
	ASCII文字コード「q」	95%	94%	ASCII文字コード「j」	82%	75%

表 6.2：ワームの ASCII 文字コード「i」

	TPR				TNR			
量子化レベル数	2	4	8	16	2	4	8	16
イントラネットA	98%	99%	98%	98%	83%	80%	86%	83%
イントラネットB	97%	98%	98%	97%	83%	80%	80%	82%

表 6.3：ファイル感染型ウイルスの「HTTP リクエスト長」

	TPR				TNR			
量子化レベル数	2	4	8	16	2	4	8	16
イントラネットA	100%	100%	100%	100%	82%	80%	81%	81%
イントラネットB	100%	100%	100%	94%	88%	86%	88%	86%

第 7 章　特徴量評価実験の考察

6 章で有効だと判断した特徴量について，正常時通信と感染時通信の重なり具合を視覚的に確認し，特徴量の有効性について考察した．また，マルウェアの種類毎の挙動を挙げ，それらの挙動について説明し，マルウェアの種類毎に有効な特徴量を考察した．

7.1　出現頻度に着目した考察

7.1.1　ワーム

7.1.1.1　ASCII 文字コードの出現頻度

6 章で求められた特徴量に対するヒストグラムを作成した．これらは横軸を出現頻度[回]，縦軸を感染時通信（正常時通信）全体のスロット数に対する当該特徴量を含んだスロットの割合[%]としている．

例えば，図 7.1 の色の濃い棒グラフは，感染時通信全体のスロットの内，ASCII 文字コード「i」の出現頻度がどれくらい含まれるのかを示した図になっている．図 7.1 の一番左の色の濃い棒グラフ（出現頻度 0…9 の色の濃い棒グラフ）は，1 スロット中に ASCII 文字コード「i」が 0～9 個含まれるスロットが、全体の 75%存在することを示している．同様に，図 7.1 の薄い棒グラフは正常時通信全体のスロットの内，ASCII 文字コード「i」の出現頻度がどれくらい含まれるのかを示した図になっている．また，図 7.2 は図 7.1 の出現頻度が 119 以降の図になっている．

5 章で有効だと判断した特徴量の中から例として，ワームの感染時通信とイントラネット A の正常時通信の ASCII 文字コード「i」のヒストグラムを図 7.1，図 7.2 に示す．また，これらは量子化レベル数を 2 としたときの結果である．

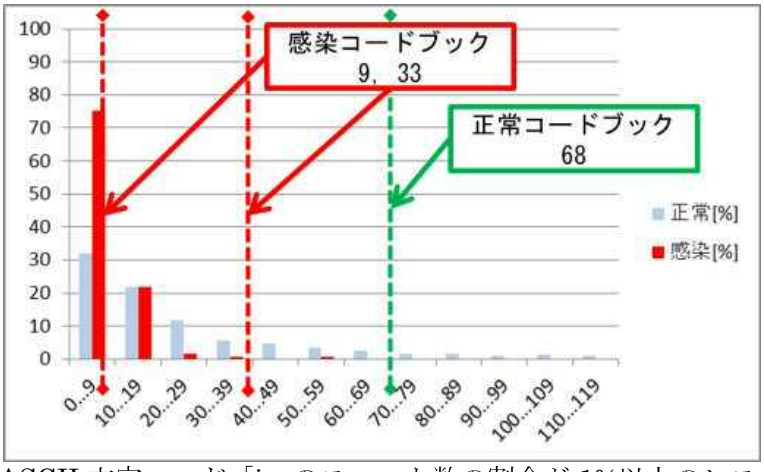

図 7.1：ASCII 文字コード「i」のスロット数の割合が 1%以上のヒストグラム
（横軸：出現頻度[回]，縦軸：スロット数の割合[%]）

図 7.2：ASCII 文字コード「i」のスロット数の割合が 1%以下のヒストグラム
（横軸：出現頻度[回]，縦軸：スロット数の割合[%]）

図 7.1，図 7.2 から，感染時通信の色の濃い棒グラフは x 軸の 59 より左側にある．つまり，感染時通信の ASCII 文字コード「i」の出現頻度が 1 スロット 59 未満である．一方正常時通信の色の薄い棒グラフは x 軸の 1,000 よりも右側にあることが確認できる．つまり，正常時通信の ASCII 文字コード「i」の出現頻度が 1 スロット 1000 以上出現することがある．そのため，感染コードブックが小さい値で，正常コードブックが大きい値で作成された．

正常時通信に ASCII 文字コード「i」が多い理由として 2 つ考えられる．第 1 の理由として，正常時通信は感染時通信よりもペイロードのバイト数が多いためである．具体的には，今回の実験データでは，正常時通信の方が感染時通信よりも，約 2 倍のバイト数があることを確認した．そのため，ASCII 文字コード「i」の出現頻度が多くなった．

第 2 の理由として，正常時通信には ASCII 文字コード「i」を含むキーワードが感染時通信よりも多く出現しているためである．例えば，正常時通信ではログイン等の個人情報を残す「Cookie」や，画像や映像などのマルチメディアを使うための「Content-type image」等の ASCII 文字コード「i」を含むキーワードが出現している．これらの ASCII 文字コード「i」を含んだ文字列と考察を以下の表 7.1 にまとめた．

表 7.1：ASCII 文字コード「i」を含んだ文字列と考察

単語	意味	考察	出現頻度 （正常/感染）
Accept-Encoding gzip	ブラウザが対応しているエンコード方式．サーバは、対応しているエンコード（gzipなど）でデータを圧縮し送信を行う	感染時通信は，インターネット接続確認などの必要最低限の情報収集を行うため，パケットの情報量が少なく，gzip圧縮を行う情報量の多い通信を行わない	58%/0% 48%/0%
Connection Keep-alive	持続的接続を行い、1セッションで複数のファイルの送受信を行う	感染時通信は，複数のリクエスト/レスポンスの発行が不必要な必要最低限の情報で行われる	62%/22% 52%/0%
Cookie	PCに個人情報を残したままにする	感染時通信は，ログイン等の個人情報を残すような通信を行わない	31%/0%
Content-type image application	ペイロード部のデータのファイルタイプを示す	感染時通信は，インターネット接続確認などを行う通信を簡単に行うため，画像があるサイトに頻繁に接続する必要がない	22%/0% 13%/0%

7.1.1.2　HTTP リクエスト長の出現頻度

ワームの感染時通信とイントラネットＡの正常時通信を用いた HTTP リクエスト長のヒストグラムを図 7.3，図 7.4 に示す．

これらの図の縦軸を感染時通信（正常時通信）全体のスロット数に対する出現頻度の割合[%]，横軸を HTTP リクエスト長としている．また，正常時通信は，上記と同様にイントラネットＡの正常時通信を用いている．

これらは量子化レベル数を 2 としたときの結果であり，図 7.3 は HTTP リクエスト長が 200 まで，図 7.4 は HTTP リクエスト長が 10,000 までを示している．また，図 7.3 から感染時データの出現頻度が 110 未満であることが確認できる．それに対し，正常時データの多くが 110 以上であることが確認できる．

正常時通信（ユーザの通信）は様々であり，HTTP リクエスト長にばらつきが存在する．感染時通信は，HTTP リクエスト長が短いものが多く，感染時コードブックが小さくなる．これにより，TPR の値が高くなったが，正常時通信の中にも HTTP リクエスト長が短くなるものもあるため，TNR の値は低くなったと考えられる．

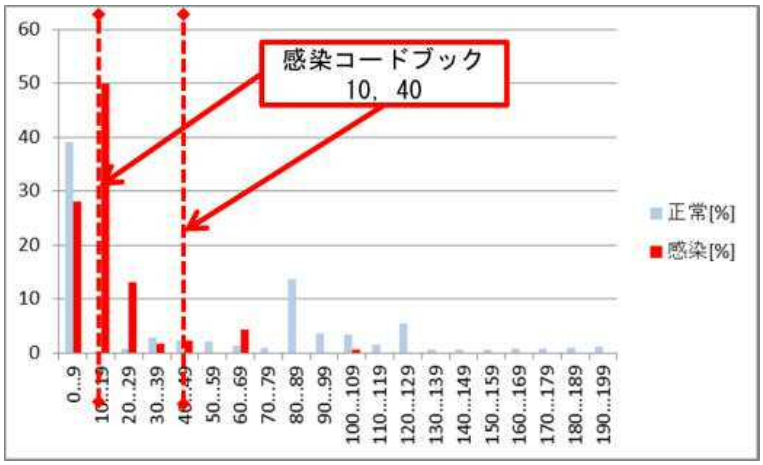

図 7.3：HTTP リクエスト長のスロット数の割合が 1%以上のヒストグラム
（横軸：リクエスト長，縦軸：スロット数の割合[%]）

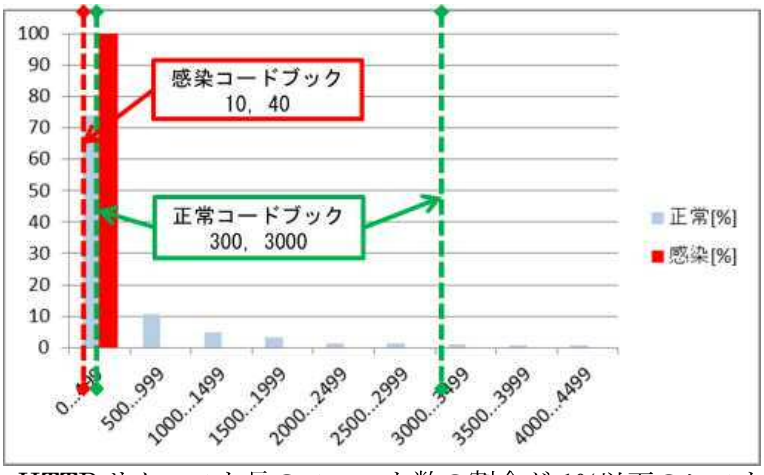

図 7.4：HTTP リクエスト長のスロット数の割合が 1%以下のヒストグラム
（横軸：リクエスト長，縦軸：スロット数の割合[%]）

7.1.2　トロイの木馬

7.1.2.1　ASCII 文字コードの出現頻度

6 章で有効だと判断した特徴量の中から例として，トロイの木馬の感染時通信とイントラネット A の正常時通信を用いた ASCII 文字コード「o」のヒストグラムを図 7.5，図 7.6 に示す．これらの図の縦軸と横軸は図 7.1，図 7.2 と同様である．また，正常時通信は上記と同様にイントラネット A の正常時通信を用いている．

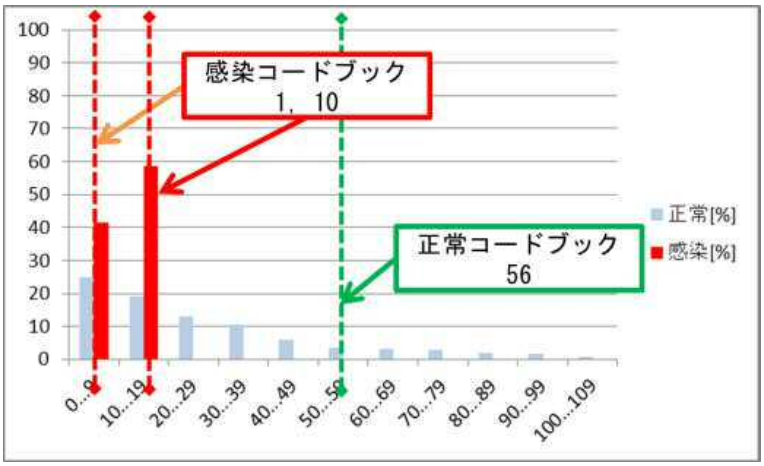

図 7.5：ASCII 文字コード「o」のスロット数の割合が 1%以上のヒストグラム
（横軸：出現頻度[回]，縦軸：スロット数の割合[%]）

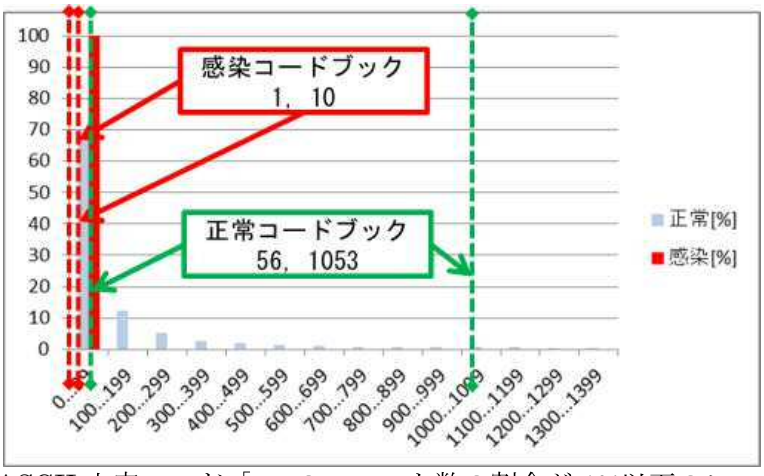

図 7.6：ASCII 文字コード「o」のスロット数の割合が 1%以下のヒストグラム
（横軸：出現頻度[回]，縦軸：スロット数の割合[%]）

図 7.5，図 7.6 から，感染時通信の ASCII 文字コード「o」の出現頻度が 20 未満までで，正常時通信の ASCII 文字コード「o」の出現頻度が 1,000 以降も出現していることが確認できる．そのため，感染コードブックが小さい値で，正常コードブックが大きい値で作成された．

正常時通信に ASCII 文字コード「o」が多い理由として 2 つ考えられる．第 1 の理由として，正常時通信は感染時通信よりもペイロードのバイト数が多いためである．具体的には，今回の実験データでは，正常時通信の方が感染時通信よりも，約 2 倍のバイト数があることを確認した．そのため，ASCII 文字コード「o」の出現頻度が多くなった．

第 2 の理由として，正常時通信には ASCII 文字コード「i」を含むキーワードが感染時通信よりも多く出現しているためである．例えば，正常時通信ではログイン等の個人情報を残す「Cookie」や，1 セッションで複数のファイルの送受信を行うための「Connection」等の ASCII 文字コード「o」を含むキーワードが出現している．これらの ASCII 文字コード「o」を含んだ文字列と考察を以下の表 7.2 にまとめた．

表 7.2：ASCII 文字コード「o」を含んだ文字列と考察

単語	意味	考察	出現頻度（正常/感染）
Accept-EncOding	ブラウザが対応しているエンコード方式．サーバは、対応しているエンコード（gzipなど）でデータを圧縮し送信を行う	感染時通信は，インターネット接続確認などの必要最低限の情報収集を行うため，パケットの情報量が少なく，gzip圧縮を行う情報量の多い通信を行わない	58%/0%
COnnectiOn	持続的接続を行い、1セッションで複数のファイルの送受信を行う	感染時通信は，複数のリクエスト/レスポンスの発行が不必要な必要最低限の情報で行われる	62%/50%
COOkie	PCに個人情報を残したままにする	感染時通信は，ログイン等の個人情報を残すような通信を行わない	31%/0%
ApplicatiOn	ペイロード部のデータのファイルタイプを示す	感染時通信は，インターネット接続確認などを行う通信を簡単に行うため，画像があるサイトに頻繁に接続する必要がない	13%/0%

7.1.2.2　HTTP リクエスト長の出現頻度

トロイの木馬の感染時通信とイントラネットAの正常時通信を用いたHTTPリクエスト長のヒストグラムを図 7.7，図 7.8 に示す．

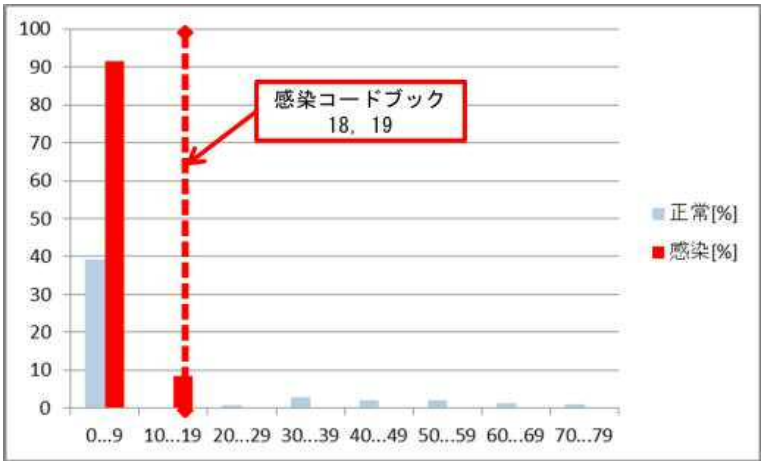

図 7.7：HTTP リクエスト長のスロット数の割合が 1%以上のヒストグラム
（横軸：リクエスト長，縦軸：スロット数の割合[%]）

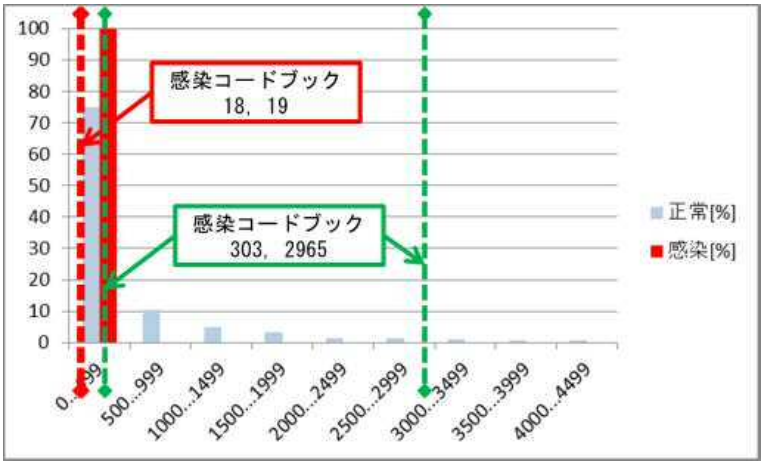

図 7.8：HTTP リクエスト長のスロット数の割合が 1%以下のヒストグラム
（横軸：リクエスト長，縦軸：スロット数の割合[%]）

これらの図の縦軸と横軸は図 7.3，図 7.4 と同様である．また，正常時通信は，上記と同様にイントラネット A の正常時通信を用いている．

これらは量子化レベル数を 2 としたときの結果であり，図 7.7 は HTTP リクエスト長が 19 まで，図 7.8 は HTTP リクエスト長が 4,500 までを示している．また，図 7.7 から感染時データの出現頻度が 20 以下であることが確認できる．それに対し，正常時データの多くが 100 以上であることが確認できる．

正常時通信（ユーザの通信）は様々であり，HTTP リクエスト長にばらつきが存在する．感染時通信は，HTTP リクエスト長が短いものが多く，感染時コードブックに分類される．これにより，TPR の値が高くなったが，正常時通信の中にも HTTP リクエスト長が短くなるものもあるため，TNR の値は低くなったと考えられる．

7.1.3　ファイル感染型ウイルス

7.1.3.1　ASCII 文字コードの出現頻度

6 章で有効だと判断した特徴量の中から例として，ファイル感染型ウイルスの感染時通信とイントラネット A の正常時通信の ASCII 文字コード「i」のヒストグラムを図 7.9，図 7.10 に示す．これらの図の縦軸と横軸は図 7.1，図 7.2 と同様である．また，正常時通信は，上記と同様にイントラネット A の正常時通信を用いている．

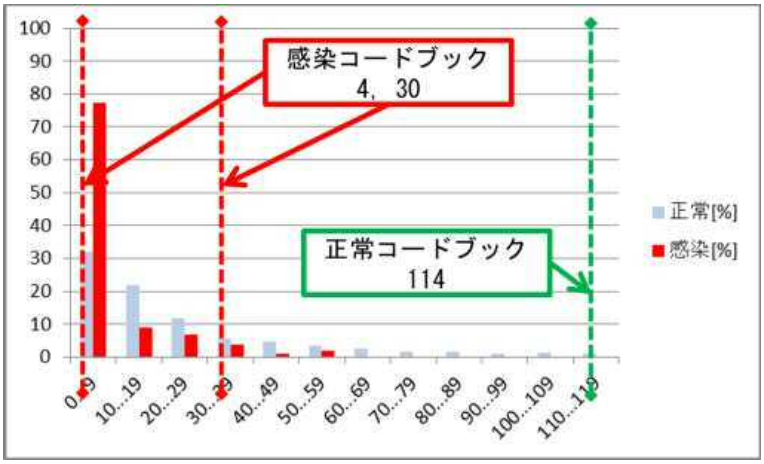

図 7.9：ASCII 文字コード「i」のスロット数の割合が 1%以上のヒストグラム
（横軸：出現頻度[回]，縦軸：スロット数の割合[%]）

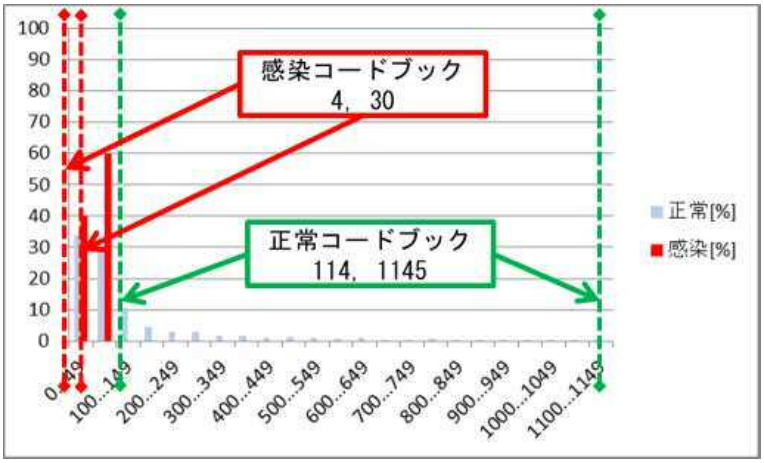

図 7.10：ASCII 文字コード「i」のスロット数の割合が 1%以下のヒストグラム
（横軸：出現頻度[回]，縦軸：スロット数の割合[%]）

図 7.9，図 7.10 から，感染時通信の ASCII 文字コード「i」の出現頻度が 59 未満までで，正常時通信の ASCII 文字コード「i」の出現頻度が 1,000 以降も出現していることが確認できる．そのため，感染コードブックが小さい値で，正常コードブックが大きい値で作成された．

正常時通信に ASCII 文字コード「i」が多い理由として 2 つ考えられる．第 1 の理由として，正常時通信は感染時通信よりもペイロードのバイト数が多いためである．具体的には，今回の実験データでは，正常時通信の方が感染時通信よりも，約 2 倍のバイト数があることを確認した．そのため，ASCII 文字コード「i」の出現頻度が多くなった．

第 2 の理由として，正常時通信には ASCII 文字コード「i」を含むキーワードが感染時通信よりも多く出現しているためである．例えば，正常時通信ではログイン等の個人情報を残す「Cookie」や，画像や映像などのマルチメディアを使うための「Content-type image」等の ASCII 文字コード「i」を含むキーワードが出現している．これらの ASCII 文字コード「i」を含んだ文字列と考察を以下の表 7.3 にまとめた．

表 7.3：ASCII 文字コード「i」を含んだ文字列と考察

単語	意味	考察	出現頻度（正常/感染）
Accept-Encoding	ブラウザが対応しているエンコード方式．サーバは、対応しているエンコード（gzipなど）でデータを圧縮し送信を行う	感染時通信は，インターネット接続確認などの必要最低限の情報収集を行うため，パケットの情報量が少なく，gzip圧縮を行う情報量の多い通信を行わない	58%/0%
Connection	持続的接続を行い、1セッションで複数のファイルの送受信を行う	感染時通信は，複数のリクエスト/レスポンスの発行が不必要な必要最低限の情報で行われる	62%/0%
Cookie	PCに個人情報を残したままにする	感染時通信は，ログイン等の個人情報を残すような通信を行わない	31%/0%
Content-type image	ペイロード部のデータのファイルタイプを示す	感染時通信は，インターネット接続確認などを行う通信を簡単に行うため，画像があるサイトに頻繁に接続する必要がない	22%/0%

7.1.3.2　HTTP リクエスト長の出現頻度

ファイル感染型ウイルスの感染時通信とイントラネット A の正常時通信を用いた HTTP リクエスト長のヒストグラムを図 7.11，図 7.12 に示す．

これらの図の縦軸と横軸は図 7.3，図 7.4 と同様である．また，正常時通信は，上記と同様にイ

ントラネット A の正常時通信を用いている．

これらは量子化レベル数を 2 としたときの結果であり，図 7.11 は HTTP リクエスト長が 110 まで，図 7.12 は HTTP リクエスト長が 3,500 までを示している．また，図 7.12 から感染時データの出現頻度が 110 以下であることが確認できる．それに対し，正常時データの多くが 110 以上であることが確認できる．

正常時通信（ユーザの通信）は様々であり，HTTP リクエスト長にばらつきが存在する．感染時データは，HTTP リクエスト長が短いものが多く，感染時コードブックに分類される．これにより，TPR の値が高くなったが正常時通信の中にも HTTP リクエスト長が短くなるものもあるため，TNR の値は低くなったと考えられる．

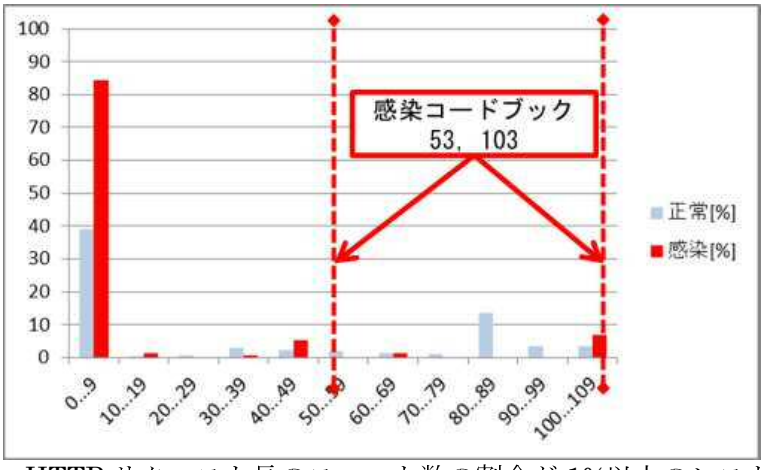

図 7.11：HTTP リクエスト長のスロット数の割合が 1%以上のヒストグラム
（横軸：リクエスト長，縦軸：スロット数の割合[%]）

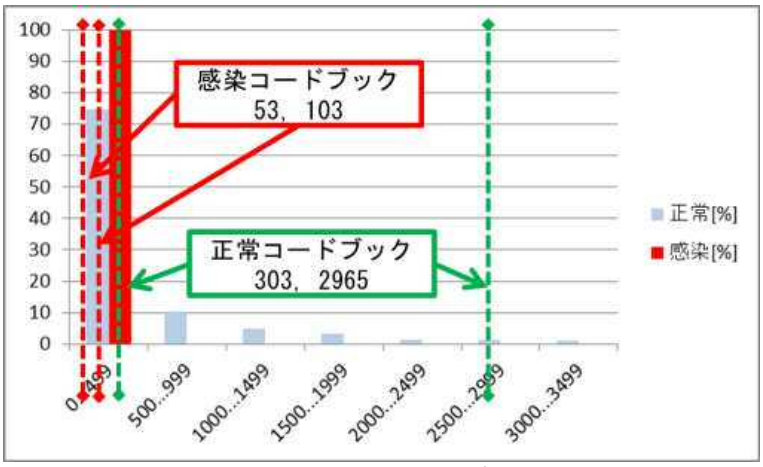

図 7.12：HTTP リクエスト長のスロット数の割合が 1%以下のヒストグラム
（横軸：リクエスト長，縦軸：スロット数の割合[%]）

7.2　出現頻度に着目した考察

マルウェア種類毎に有効な特徴量について，ワーム，トロイの木馬，ファイル感染型ウイルスに分類されるマルウェアそれぞれにおいて通信の中に含まれるペイロード情報を挙げ，それらの挙動を説明し，有効な特徴量について考察した．

7.2.1　ワーム

インターネット接続確認

ワームが感染活動を行い，PC を感染させた後に，その PC がインターネットに接続されているかを確認する．その際ワームは，特定のドメインにインターネット接続確認を行う．特定のドメインとは，「*www.whatismyipaddress.com*」や「*checkip.dyndns.org*」のような IP アドレスを表示するサイトである．

攻撃通信を行うためのマルウェアのダウンロード

起点となるワームに感染後，各サーバから他のマルウェアを HTTP GET によりダウンロードする．このコマンドは，「*GET /vss.exe HTTP/1.0*」や「*GET /fdc1.data HTTP/1.0*」のようなものである．感染後に行う通信は，図 7.3，図 7.4 より，HTTP リクエスト長が概ね 110 未満と短くなる．それに対して，正常時通信は概ね 110 以上の通信を行っているものがほとんど（スロット数の割合が 95%以上）である．

よって，ドメイン等に含まれる ASCII 文字コードが正常時通信と比べて出現頻度が低いことや，HTTP リクエスト長が感染時通信で短くなることから，表 6.1 で示した特徴量（ASCII 文字コード「f」等）はマルウェア感染検知に有効であると判断できる．

7.2.2 トロイの木馬

攻撃通信を行うためのマルウェアのダウンロード

起点となるマルウェアをダウンロード後，各サーバから他のマルウェアを HTTP GET によりダウンロードする．例としては，「*GET /vot.exe HTTP/1.0*」や「*GET /15Psv3zJ/4ah6NuS.exe HTTP/1.0*」のような HTTP GET によるダウンロードを行う．HTTP GET による通信だけを行っているものが多く，ペイロードの情報量が少ない．そのため，正常時通信と比較すると、改行（*¥r¥n*）の数が少ない傾向がある．また感染後に行う通信は，図 7.7，図 7.8 より，HTTP リクエスト長が概ね 20 未満と短くなる．それに対し，正常時通信は概ね 110 以上であるものが大半（スロット数の割合が 95%以上）である．

よって HTTP GET 等に含まれる ASCII 文字コードが正常時通信と比べて出現頻度が低いことや，HTTP リクエスト長が感染時通信で短くなることから，表 6.1 に示した特徴量（ASCII 文字コード「e」等）はマルウェア感染検知に有効であると判断できる．

7.2.3 ファイル感染型ウイルス

IRC 通信による C&C サーバに接続（IRC 接続）

ファイル感染型ウイルスは感染活動を行うための準備として，IRC 通信を行い C&C サーバに接続する．IRC 接続を行った後，攻撃通信を行うためのマルウェアのダウンロードや標的に対して妨害攻撃を行う等の活動が行われる．今回用いた検体では，攻撃通信を行うためのマルウェアのダウンロードを行っていた．IRC 通信を行う際の通信内容は特定の文字列（IRC ドメイン *:norks.org 001* 等）が単位スロット毎に同程度の数が繰り返し多く出現する．具体的には，IRC 通信に含まれる ASCII 文字コードの出現頻度は，正常時通信ではバラバラであるが， 感染時通信では 60～75 で一定になっていた．

攻撃通信を行うためのマルウェアのダウンロード

起点となるマルウェアをダウンロード後，マルウェアは各サーバから他のマルウェアを HTTP GET によりダウンロードする．例として，「*GET /jiri.data HTTP/1.0*」や「*GET 44.data HTTP/1.0*」などのような HTTP GET によるダウンロードを行う．また感染後に行う通信は，図 12，図 13 より，HTTP リクエスト長が 110 未満であるのに対し，正常時通信は概ね 110 以上であるものが大半（スロット数の割合が 95%以上）である．

よって，IRC 通信に含まれる ASCII 文字コードが正常時通信と比べて出現頻度が同程度の数で繰り返し多い（正常時通信ではバラバラであるが，感染時通信では 60～75 で一定になっていた）ことや，HTTP リクエスト長が感染時通信で短くなることから，表 6.1 で示した特徴量（ASCII 文字コード「d」等）はマルウェア感染検知に有効であると判断できる．

以上マルウェア 3 種の考察から，今回用いたデータセットの場合，HTTP 通信を用いたマルウェアの感染活動は，インターネットの接続確認や攻撃通信を行うためのマルウェアのダウンロードを行っている．また，攻撃通信を行うためのマルウェアのダウンロードを行う際に，IRC 通信を行うものや外部サーバから直接ダウンロードを行うもの等，マルウェアの種類毎で異なる挙動が確認できた．さらに，感染活動を行うときのマルウェアの種類毎でマルウェアのダウンロードを行う時等のペイロード情報の文字列が異なるため，マルウェアの種類毎で有効だと判断された特徴量が異なったと考えられる．

第 8 章　有効な特徴量を組み合わせた検知システムの実験・考察

正常時通信と感染時通信をより正確に分類するマルウェア感染検知システムの実装をするために，有効な特徴量の組み合わせの評価実験を行った．具体的には，7 章で有効だと判断された 20 個の特徴量を用い，最適な組み合わせを評価した．その際，2 つの特徴量の組み合わせ，3 つの特徴量の組み合わせを評価対象とした．

8.1　実験概要

正常時通信と感染時通信をより正確に分類するためには，正常時通信と感染時通信を分類する境界線を評価するだけでなく，感染時通信と正常時通信と誤検知した感染時通信を分類する境界線，正常時通信と感染時通信と誤検知した正常時通信を分類する境界線を評価する必要がある．

そのため本研究では，これらの 3 箇所の境界線を作成・評価するために，ロジスティック回帰分析を用い，評価実験を行った．

8.1.1　ロジスティック回帰分析

ロジスティック回帰分析は，1 つのカテゴリカル変数を目的変数とし，その目的変数をその他の変数で説明する形のモデルを使って分析する手法である[24]．まず，目的変数は 2 値の場合を考える．Yを 0 または 1 の値をとる変数とする．このとき，Yの分布は

$$P(Y=1)=p,\ P(Y=0)=1-p \tag{8.1}$$

と表される．ここでは単に p を使って表しているが，p は説明変数 x の値によって変化するものとし，ロジスティック回帰モデルでは，

$$\log\frac{p}{1-p}=a+bx \tag{8.2}$$

と仮定する．この式を変形する事で，pはxの関数として

$$p=\frac{exp(a+bx)}{1+exp(a+bx)} \tag{8.3}$$

と表す事ができる．今，$(x,\ X)$が n 回観測されたものとし，i番目のデータのxの値をx_i，Yの値をy_iとすると，

$$P(Y=1)=\frac{exp(a+bx_i)}{1+exp(a+bx_i)} \qquad P(Y=1)=\frac{1}{1+exp(a+bx_i)} \tag{8.4}$$

となることから，

$$P(Y=y_i)=\frac{\{exp(a+bx_i)\}^{y_i}}{1+exp(a+bx_i)} \tag{8.5}$$

のように，y_iを使って 1 つの表現にまとめることができる．n 回の測定結果が互いに独立であると仮定すると，n 回分の結果を生じる確率は，それぞれの結果が生じる確率の積で表されるので，

$$\prod_{i=1}^{n}\frac{\{exp(a+bx_i)\}^{y_i}}{1+exp(a+bx_i)} \tag{8.6}$$

となる．この確率を最大にするa及びbの値をa,bの推定量として用いる．この方法を最尤法と呼ばれており，そのときのa,bの推定量を最尤推定量と呼ぶ．実際には，対数をとったものを考えるほ

うが扱いやすいため

$$l(a,b) = \sum_{i=1}^{n} y_i(a + bx_i) - \sum_{i=1}^{n} \log\{1 + exp(a + bx_i)\} \tag{8.7}$$

を最大にする方法が取られる．

また，学習結果のωと分類したい事例 x の内積を計算し，その正負でクラスの判別を行う．具体的に内積計算の式を展開すると，

$$\omega^T x = (\omega_1, \omega_2)\begin{pmatrix} f_1 \\ f_2 \end{pmatrix} = \omega_0 + \omega_1 f_1 + \omega_2 f_2 > 0 \tag{8.8}$$

さらに変形して，

$$f_2 > -\frac{\omega_1}{\omega_2} f_1 - \frac{\omega_0}{\omega_2} \tag{8.9}$$

式(8.9)は直線で領域を分割することを表す．この直線がクラス 1 とクラス 2 の境界線となる．

本研究では，ロジスティック回帰分析を行う機械学習ツール LIBLINEAR を用いた．LIBLINEAR とは，大規模なデータを扱うことができるオープンソースの線形クラスタリングライブラリである．また，ロジスティック回帰分析と SVM をサポートし，コマンドラインでの利用とプログラムからの呼び出しが可能である[25]．

この LIBLINEAR を用い，ロジスティックモデルを作成した．このとき，クラスとして正常時通信を 1，感染時通信と誤検知した正常時通信を 2，感染時通信を-1，正常時通信と誤検知した感染時通信-2 とした．また，要素を組み合わせに選択した特徴量のスコア（コードブックとテストデータの最小距離）とした．

8.1.2 実験に用いるペイロードの特徴量

7 章で有効だと判断された 20 個の特徴量を用いる（表 8.1）．

表 8.1：7 章で有効だと判断された 20 個の特徴量

7章で有効だと判断された特徴量	
HTTPリクエスト長	ASCII文字コード「S」
ASCII文字コード「NL*」	ASCII文字コード「d」
ASCII文字コード「CR*」	ASCII文字コード「e」
ASCII文字コード「0」	ASCII文字コード「f」
ASCII文字コード「1」	ASCII文字コード「i」
ASCII文字コード「2」	ASCII文字コード「o」
ASCII文字コード「5」	ASCII文字コード「p」
ASCII文字コード「A」	ASCII文字コード「r」
ASCII文字コード「C」	ASCII文字コード「t」
ASCII文字コード「J」	ASCII文字コード「x」

8.1.3 実験データ

正常時通信にあるイントラネットに流れる通信データを用い，感染時通信に CCCDATAset2009, 2010, 2011（以下 CCC2009, CCC2010, CCC2011），D3M2012, 2013, PRACTICE2013 を用いた（表 8.2）[22].

表 8.2：各データセットのマルウェア

種類	CCCDATAset2011	CCCDATAset2010	CCCDATAset2009	D3M2012	PRACTICE2013	D3M2013
ワーム	WORM_DOWNAD.AD	WORM_DOWNAD.AD	WORM_SWTYMLAI.CD		WORM_MYTOB.IR	
		WORM_MAINBOT.AH				
		WORM_MAINBOT.FY				
		WORM_PALEVO.SMD				
		WORM_PALEVO.BL				
トロイの木馬			TROJ_BUZUS.AGB	TROJ_GEN.R47C7C2	PWS-Zbot.gen.alu	Trojan-FakeAV.Win32.SecurityShield.rkz
				Trojan.Generic.KD.578000	TROJ_GEN.RCCC7IT	Trojan-Ransom.Win32.PornoAsset.aeuz
				Trojan.Generic.KD.410743	TROJ_GEN.USBJ05ACN	HEUR:Trojan.Win32.Bublik.ahjt
				Trojan-Dropper.Win32.Dapato.aoxn	Trojan-Spy.Win32.SpyEyes.wb	Backdoor.Win32.Pmax.qgu
						Backdoor.Win32.Pmax.qgv
						Backdoor.Win32.Zaccess.bkrm
						Backdoor.Win32.Pmax.qgs
						Backdoor.Win32.Pmax.qgr
						Trojan-Dropper.Win32.Agent.btsn
ファイル感染型ウイルス		PE_VIRUT.AV	PE_BOBAX.AK			
		PE_VIRUT.XV				

これらを正常/感染コードブック作成用の学習データ，テストデータ用，マッチング対象データ用の 3 つに年度毎で分割した．これは，年度の古いマルウェアで学習し，年度の新しいマルウェア（未知のマルウェア）が検知できるかを確認するためである．以下に正常時通信と感染時通信を年度毎で分類した表をまとめた（表 8.3, 8.4）．

表 8.3：感染時通信を年度毎で分類したまとめ

グループ	マッチング対象データ用（2013年）	テストデータ用（2012年+2011年）	コードブック作成用の学習データ（2010年+2009年）
マルウェア検体名	WORM_MYTOB.IR	Trojan.Generic.KD.578000	WORM_DOWNAD.AD_2010
	PWS-Zbot.gen.alu	Trojan.Generic.KD.410743	WORM_MAINBOT.FY
	TROJ_GEN.RCCC7IT	TROJ_GEN.R47C7C2	WORM_PALEVO.SMD
	TROJ_GEN.USBJ05ACN	Trojan-Dropper.Win32.Dapato.aoxn	WORM_MAINBOT.AH
	Trojan-Spy.Win32.SpyEyes.wb	WORM_DOWNAD.AD_2011	WORM_SWTYMLAI.CD
	Trojan-FakeAV.Win32.SecurityShield.rkz		WORM_PALEVO.BL
	Trojan-Ransom.Win32.PornoAsset.aeuz		PE_VIRUT.AV
	HEUR:Trojan.Win32.Bublik.ahjt		PE_VIRUT.XV
	Backdoor.Win32.Pmax.qgu		TROJ_BUZUS.AGB
	Backdoor.Win32.Pmax.qgv		PE_BOBAX.AK
	Backdoor.Win32.Zaccess.bkrm		
	Backdoor.Win32.Pmax.qgs		
	Backdoor.Win32.Pmax.qgr		
	Trojan-Dropper.Win32.Agent.btsn		
総スロット数	1994	880	487

表 8.4：正常時通信を年度毎で分類したまとめ

グループ	マッチング対象データ用（2013年）	テストデータ用（2012年+2011年）	コードブック作成用の学習データ（2010+2009）
総スロット数	2000	4000	4000

8.1.4 評価方法

有効な特徴量の組み合わせを評価する感染時通信と正常時通信の識別方法について説明する．

8.1.4.1 ベクトル量子化によるコードブック作成

ベクトル量子化を用いて，感染時通信のみを用いて学習を行った感染コードブックと，正常時通信のみを用いて学習を行った正常コードブックを予め作成する．今回は，各特徴量を個別に評価することが目的であるため，特徴量毎に 1 次元コードブックを作成した．ベクトル量子化のアルゴリズムには，LBG+Splitting アルゴリズムを用い，レベル数は 2，4，8，16 の 4 種類とした．

8.1.4.2　コードブックを用いた計算方法

ベクトル量子化を用いて作成したコードブックを用い，テストデータと感染，正常コードブックとの距離を計算し，感染(正常)コードブックとの距離の方が小さければ感染(正常)と識別した．

8.1.4.3　スコアの計算方法

正常テストデータを用いたときの正常/感染コードブックとの最小距離，感染テストデータを用いたときの正常/感染コードブックとの最小距離をスコアとする．

今回の実験では，以下のようなスコアとした．

① 感染テストデータと感染コードブックとのスコア（感染-感染のスコア（評価：-1））
② 感染テストデータと正常コードブックとのスコア（感染-正常のスコア（評価：-2））
③ 正常テストデータと正常コードブックとのスコア（正常-正常のスコア（評価：1））
④ 正常テストデータと感染コードブックとのスコア（正常-感染のスコア（評価：2））

8.1.4.4　ロジスティック回帰分析を用いた識別率の計算方法

LIBLINEAR を用いて作成した学習用ロジスティックモデルに対し，正常時通信と感染時通信が混在したマッチング用の対象データが，正常と感染に正しく分類しているかの指標を識別率としている．

$$識別率 = \frac{正常と感染を正しく分類した数}{全スロット数} \tag{8.10}$$

本実験では，正常と感染を識別する有効な特徴量の識別率の指標を 95%として評価を行った．実験概要，8.1.4.3 のスコアから，識別率を評価する境界線として以下のものである（図 8.1）．

(1) ①と③の境界線（正常時通信と感染時通信を分類する境界線）
(2) ③と④の境界線（正常時通信を感染時通信と誤検知した正常時通信を分類する境界線）
(3) ①と②の境界線（感染時通信を正常時通信と誤検知した感染時通信を分類する境界線）

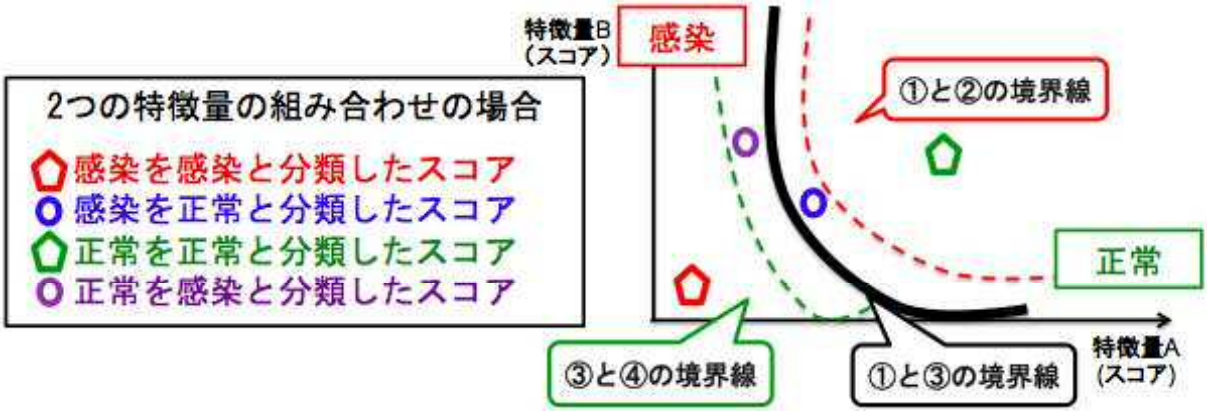

図 8.1：評価する 3 つの境界線

8.2　実験結果

8.2.1　2 つの特徴量の組み合わせの結果

8.1.2 節で述べた 20 個の特徴量から，2 つの有効な特徴量を組み合わせた評価実験の結果を以下の表に示す．

表 8.5：量子化レベル毎の有効な特徴量の組み合わせの数

量子化レベル	2	4	8	16
①と③の境界線	54	28	45	48
①と②の境界線	23	40	90	22
③と④の境界線	27	0	0	0

有効な特徴量を組み合わせることで，①と③の境界線では，量子化レベル 2 で 54 個，量子化レベル 4 で 28 個，量子化レベル 8 で 45 個，量子化レベル 16 で 48 個の有効な特徴量の組み合わせが，識別率 95%以上になった．また，①と②の境界線では，量子化レベル 2 で 23 個，量子化レベル 4 で 40 個，量子化レベル 8 で 90 個，量子化レベル 16 で 22 個の有効な特徴量の組み合わせが，識別率 95%以上になった．③と④の境界線では，量子化レベル 2 で 27 個の有効な特徴量の組み合わせが，識別率 95%以上になった．

さらに，7 章で有効だと判断した特徴量の識別率よりも識別率が高くなった特徴量の組み合わせを以下の表に示す．

表 8.6.1：7 章で評価した特徴量よりも識別率が高くなった 2 つの有効な特徴量の組み合わせ

評価した境界線：①と③の境界線　組み合わせ数：2

特徴量	特徴量の組み合わせ	識別率(%)	量子化レベル数
ASCII文字コード「0」(TPR：97.5%)	ASCII文字コード「NL*」 ASCII文字コード「0」	100	8
	ASCII文字コード「0」 ASCII文字コード「o」	100	16
	ASCII文字コード「0」 ASCII文字コード「r」	100	16
	ASCII文字コード「0」 ASCII文字コード「t」	100	16
ASCII文字コード「e」(TPR：99.0%)	ASCII文字コード「NL*」 ASCII文字コード「e」	100	8,16
	ASCII文字コード「e」 ASCII文字コード「o」	100	16
	ASCII文字コード「e」 ASCII文字コード「r」	100	16
	ASCII文字コード「e」 ASCII文字コード「t」	100	16
ASCII文字コード「i」(TPR：98.5%)(TNR：83.0%)	ASCII文字コード「NL*」 ASCII文字コード「i」	100	8
	ASCII文字コード「i」 ASCII文字コード「r」	100	16
ASCII文字コード「o」(TPR：97 .0%)	HTTPリクエストの長さ ASCII文字コード「o」	97.5	2
	ASCII文字コード「C」 ASCII文字コード「o」	97.3	2
	ASCII文字コード「S」 ASCII文字コード「o」	97.5	2
	ASCII文字コード「d」 ASCII文字コード「o」	97.3	2
	ASCII文字コード「2」 ASCII文字コード「o」	97.7	4
	ASCII文字コード「A」 ASCII文字コード「o」	97.8	4
		100	16
	ASCII文字コード「NL*」 ASCII文字コード「o」	100	16
	ASCII文字コード「0」 ASCII文字コード「o」	100	16
	ASCII文字コード「1」 ASCII文字コード「o」	100	16
	ASCII文字コード「5」 ASCII文字コード「o」	100	16
	ASCII文字コード「e」 ASCII文字コード「o」	100	16
	ASCII文字コード「o」 ASCII文字コード「p」	100.0	16
	ASCII文字コード「o」 ASCII文字コード「t」	100	16
	ASCII文字コード「o」 ASCII文字コード「v」	98	16
ASCII文字コード「p」(TPR：99.0%)	ASCII文字コード「NL*」 ASCII文字コード「p」	97.7	2
		100	16
	ASCII文字コード「o」 ASCII文字コード「p」	100	16
	ASCII文字コード「p」 ASCII文字コード「r」	100	16
	ASCII文字コード「p」 ASCII文字コード「t」	100	16

評価した境界線：③と④の境界線　組み合わせ数：2

特徴量	特徴量の組み合わせ	識別率(%)	量子化レベル数
ASCII文字コード「0」(TPR：97.5%)	ASCII文字コード「0」 ASCII文字コード「A」	100	2
	ASCII文字コード「0」 ASCII文字コード「r」	100	2
ASCII文字コード「e」(TPR：99.0%)	ASCII文字コード「e」 ASCII文字コード「r」	100	2
ASCII文字コード「f」(TPR：97.5%)	ASCII文字コード「f」 ASCII文字コード「r」	100	2
ASCII文字コード「i」(TPR：98.5%)(TNR：83.0%)	ASCII文字コード「A」 ASCII文字コード「i」	100	2
	ASCII文字コード「i」 ASCII文字コード「r」	100	2
ASCII文字コード「o」(TPR：97 .0%)	ASCII文字コード「A」 ASCII文字コード「o」	100	2
	ASCII文字コード「o」 ASCII文字コード「r」	100	2
ASCII文字コード「p」(TPR：97 .0%)	ASCII文字コード「A」 ASCII文字コード「p」	100	2

表 8.6.2：7 章で評価した特徴量よりも識別率が高くなった 2 つの有効な特徴量の組み合わせ

評価した境界線:①と②の境界線 組み合わせ数:2

特徴量	特徴量の組み合わせ	識別率(%)	量子化レベル数	特徴量	特徴量の組み合わせ	識別率(%)	量子化レベル数	特徴量	特徴量の組み合わせ	識別率(%)	量子化レベル数	特徴量	特徴量の組み合わせ	識別率(%)	量子化レベル数
ASCII文字コード「0」(TPR：97.5%)	ASCII文字コード「0」ASCII文字コード「c」	98.9	16	ASCII文字コード「e」(TPR：99.0%)	ASCII文字コード「NL*」ASCII文字コード「e」	99.5	2	ASCII文字コード「j」(TPR：97.5%)	ASCII文字コード「CR」ASCII文字コード「f」	99.5	8	ASCII文字コード「o」(TPR：97.0%)	ASCII文字コード「NL*」ASCII文字コード「o」	99.1	2
	ASCII文字コード「0」ASCII文字コード「i」	99.4	16		ASCII文字コード「CR」ASCII文字コード「e」	99.5	2		ASCII文字コード「2」ASCII文字コード「f」	99.5	8			100	16
ASCII文字コード「S」(TPR：98.0%)	ASCII文字コード「NL*」ASCII文字コード「S」	100	4			99.4	8		ASCII文字コード「5」ASCII文字コード「f」	99.5	8		ASCII文字コード「CR」ASCII文字コード「o」	99.3	2
	ASCII文字コード「CR」ASCII文字コード「S」	99.9	4		ASCII文字コード「d」ASCII文字コード「e」	100	2		ASCII文字コード「A」ASCII文字コード「f」	99.8	8			98	4
		100	8			99.5	4		ASCII文字コード「C」ASCII文字コード「f」	99.5	8			99.5	8
	ASCII文字コード「2」ASCII文字コード「S」	100	4			99.6	8		ASCII文字コード「S」ASCII文字コード「f」	100	8		ASCII文字コード「d」ASCII文字コード「o」	100	2
		100	8		ASCII文字コード「e」ASCII文字コード「i」	99.5	2		ASCII文字コード「d」ASCII文字コード「f」	99.5	8			99.4	4
	ASCII文字コード「C」ASCII文字コード「S」	100	4		ASCII文字コード「e」ASCII文字コード「r」	99.5	2		ASCII文字コード「e」ASCII文字コード「f」	99.5	8			99.5	8
		99.9	8			99.6	4		ASCII文字コード「f」ASCII文字コード「i」	99.6	8		ASCII文字コード「i」ASCII文字コード「o」	99.5	2
	ASCII文字コード「S」ASCII文字コード「d」	100	4		ASCII文字コード「e」ASCII文字コード「t」	99.5	2		ASCII文字コード「f」ASCII文字コード「o」	99.5	8		ASCII文字コード「o」ASCII文字コード「t」	99.5	2
		100	8			99.4	8		ASCII文字コード「f」ASCII文字コード「p」	99.6	8			99.7	4
	ASCII文字コード「S」ASCII文字コード「e」	100	4			100	16		ASCII文字コード「f」ASCII文字コード「t」	99.5	8			99.3	8
	ASCII文字コード「S」ASCII文字コード「o」	99.1	4		ASCII文字コード「2」ASCII文字コード「e」	100	4		ASCII文字コード「f」ASCII文字コード「v」	99.5	8		ASCII文字コード「2」ASCII文字コード「o」	100	4
	ASCII文字コード「S」ASCII文字コード「r」	100	4			99.5	8	ASCII文字コード「i」(TPR：98.5%)(TNR：83.0%)	ASCII文字コード「NL*」ASCII文字コード「i」	99.6	2			99.5	8
	ASCII文字コード「S」ASCII文字コード「t」	99.8	4		ASCII文字コード「C」ASCII文字コード「e」	100	4		ASCII文字コード「CR」ASCII文字コード「i」	99.4	2		ASCII文字コード「C」ASCII文字コード「o」	99.9	4
	ASCII文字コード「5」ASCII文字コード「S」	99.8	8			99.1	8			99.5	8			99.1	8
	ASCII文字コード「S」ASCII文字コード「f」	100	8			99	16			99.4	16		ASCII文字コード「S」ASCII文字コード「o」	99.1	4
	ASCII文字コード「S」ASCII文字コード「i」	100	8		ASCII文字コード「S」ASCII文字コード「e」	100	4		ASCII文字コード「d」ASCII文字コード「i」	100	2			97.4	8
	ASCII文字コード「S」ASCII文字コード「p」	99.9	8		ASCII文字コード「e」ASCII文字コード「f」	99.5	4			99.6	8		ASCII文字コード「e」ASCII文字コード「o」	99.5	4
					ASCII文字コード「e」ASCII文字コード「o」	99.5	4		ASCII文字コード「e」ASCII文字コード「i」	99.5	2			99.5	8
						99.5	8		ASCII文字コード「i」ASCII文字コード「o」	99.5	2		ASCII文字コード「o」ASCII文字コード「r」	99.3	4
					ASCII文字コード「e」ASCII文字コード「p」	99.5	8		ASCII文字コード「i」ASCII文字コード「t」	99.5	2		ASCII文字コード「A」ASCII文字コード「o」	99.5	8
					ASCII文字コード「e」ASCII文字コード「v」	99.4	8		ASCII文字コード「2」ASCII文字コード「i」	99.5	8		ASCII文字コード「f」ASCII文字コード「o」	99.5	8
										99.4	16		ASCII文字コード「o」ASCII文字コード「p」	99	8
									ASCII文字コード「5」ASCII文字コード「i」	99.5	8		ASCII文字コード「o」ASCII文字コード「v」	99.5	8
										99.5	16	ASCII文字コード「p」(TPR：97.0%)	ASCII文字コード「CR」ASCII文字コード「p」	99.5	8
									ASCII文字コード「A」ASCII文字コード「i」	99.5	8		ASCII文字コード「2」ASCII文字コード「p」	99.5	8
									ASCII文字コード「C」ASCII文字コード「i」	99.5	8		ASCII文字コード「5」ASCII文字コード「p」	99.3	8
										98.9	16		ASCII文字コード「C」ASCII文字コード「p」	99.2	8
									ASCII文字コード「S」ASCII文字コード「i」	100	8		ASCII文字コード「S」ASCII文字コード「p」	99.9	8
									ASCII文字コード「f」ASCII文字コード「i」	99.6	8		ASCII文字コード「d」ASCII文字コード「p」	99.7	8
									ASCII文字コード「i」ASCII文字コード「p」	99.6	8		ASCII文字コード「e」ASCII文字コード「p」	99.5	8
									ASCII文字コード「i」ASCII文字コード「v」	99.5	8		ASCII文字コード「f」ASCII文字コード「p」	99.6	8
										99.4	16		ASCII文字コード「i」ASCII文字コード「p」	99.5	8
									ASCII文字コード「0」ASCII文字コード「i」	99.4	16		ASCII文字コード「p」ASCII文字コード「v」	99.1	8

表 8.6.1，8.6.2 より，以下の有効な特徴量の組み合わせが， 7 章で有効だと判断した特徴量の識別率よりも，識別率が高くなった.

- ①と③の境界線：
  ASCII 文字コード「0」で 4 つ，ASCII 文字コード「e」で 4 つ，ASCII 文字コード「i」で 2 つ, ASCII 文字コード「0」で 4 つ, ASCII 文字コード「e」で 4 つ, ASCII 文字コード「p」で 4 つ
- ①と②の境界線：
  ASCII 文字コード「0」で 2 つ，ASCII 文字コード「S」で 13 つ，ASCII 文字コード「e」で 13 つ，ASCII 文字コード「f」で 13 つ，ASCII 文字コード「i」で 15 つ，ASCII 文字コード「o」で 14 つ， ASCII 文字コード「p」で 10 つ
- ③と④の境界線：
  ASCII 文字コード「0」で 2 つ， ASCII 文字コード「e」で 1 つ，ASCII 文字コード「f」で 1 つ，ASCII 文字コード「i」で 2 つ，ASCII 文字コード「o」で 2 つ， ASCII 文字コード「p」で 1 つ

さらにその中でも，2 つの特徴量の識別率が共に向上した組み合わせが以下の通りになった.

- ①と③の境界線：3 つ
  - ASCII 文字コード「0」と ASCII 文字コード「o」：
    （ASCII 文字コード「0」: 97.5% → 100%，ASCII 文字コード「o」: 97.0% → 100%）
  - ASCII 文字コード「e」と ASCII 文字コード「o」：
    （ASCII 文字コード「e」: 99.0% → 100%，ASCII 文字コード「o」: 97.0% → 100%）
  - ASCII 文字コード「o」と ASCII 文字コード「p」：
    （ASCII 文字コード「o」: 97.0% → 100%，ASCII 文字コード「p」: 99.0% → 100%）
- ①と②の境界線：17 つ
  - ASCII 文字コード「0」と ASCII 文字コード「i」：
    （ASCII 文字コード「0」: 97.5% → 99.4%，ASCII 文字コード「i」: 98.5% → 99.4%）
  - ASCII 文字コード「S」と ASCII 文字コード「e」：
    （ASCII 文字コード「S」: 98.0% → 100%，ASCII 文字コード「e」: 99.0% → 100%）
  - ASCII 文字コード「S」と ASCII 文字コード「o」：
    （ASCII 文字コード「S」: 98.0% → 99.1%，ASCII 文字コード「o」: 97.0% → 99.1%）
  - ASCII 文字コード「S」と ASCII 文字コード「f」：
    （ASCII 文字コード「S」: 98.0% → 100%，ASCII 文字コード「f」: 98.5% → 100%）
  - ASCII 文字コード「S」と ASCII 文字コード「i」：
    （ASCII 文字コード「S」: 98.0% → 100%，ASCII 文字コード「i」: 98.5% → 100%）
  - ASCII 文字コード「S」と ASCII 文字コード「p」：
    （ASCII 文字コード「S」: 98.0% → 99.9%，ASCII 文字コード「p」: 99.0% → 99.9%）
  - ASCII 文字コード「e」と ASCII 文字コード「i」：
    （ASCII 文字コード「e」: 99.0% → 99.5%，ASCII 文字コード「i」: 98.5% → 99.5%）
  - ASCII 文字コード「S」と ASCII 文字コード「e」：
    （ASCII 文字コード「S」: 98.0% → 100%，ASCII 文字コード「e」: 99.0% → 100%）
  - ASCII 文字コード「e」と ASCII 文字コード「f」：
    （ASCII 文字コード「e」: 99.0% → 99.5%，ASCII 文字コード「f」: 98.5% → 99.5%）
  - ASCII 文字コード「e」と ASCII 文字コード「o」：
    （ASCII 文字コード「e」: 99.0% → 99.5%，ASCII 文字コード「o」: 97.0% → 99.5%）
  - ASCII 文字コード「e」と ASCII 文字コード「p」：
    （ASCII 文字コード「e」: 99.0% → 99.5%，ASCII 文字コード「p」: 99.0% → 99.5%）
  - ASCII 文字コード「f」と ASCII 文字コード「i」：
    （ASCII 文字コード「f」: 98.5% → 99.6%，ASCII 文字コード「i」: 98.5% → 99.6%）
  - ASCII 文字コード「f」と ASCII 文字コード「o」：
    （ASCII 文字コード「f」: 98.5% → 99.5%，ASCII 文字コード「o」: 97.0% → 99.5%）
  - ASCII 文字コード「f」と ASCII 文字コード「p」：

（ASCII 文字コード「f」:98.5% → 99.6%，ASCII 文字コード「p」:99.0% → 99.6%）

- ASCII 文字コード「i」と ASCII 文字コード「o」：
  （ASCII 文字コード「i」:98.5% → 99.6%，ASCII 文字コード「o」:97.0% → 99.6%）
- ASCII 文字コード「i」と ASCII 文字コード「p」：
  （ASCII 文字コード「i」:98.5% → 99.6%，ASCII 文字コード「p」:99.0% → 99.6%）
- ASCII 文字コード「o」と ASCII 文字コード「p」：
  （ASCII 文字コード「o」:97.0% → 99.0%，ASCII 文字コード「p」:97.0% → 99.0%）

8.2.2　3 つの特徴量の組み合わせの結果

8.1.2 節で述べた 20 個の特徴量から，3 つの有効な特徴量を組み合わせた評価実験の結果を以下の表に示す

表 8.7：量子化レベル毎の有効な特徴量の組み合わせの数

量子化レベル	2	4	8	16
①と③の境界線	71	1	6	2
①と②の境界線	16	8	68	1
③と④の境界線	0	0	0	0

有効な特徴量を組み合わせることで，①と③の境界線では，量子化レベル 2 で 71 個，量子化レベル 4 で 1 個，量子化レベル 8 で 6 個，量子化レベル 16 で 2 個の有効な特徴量の組み合わせが，識別率 95%以上になった．また，①と②の境界線では，量子化レベル 2 で 16 個，量子化レベル 4 で 8 個，量子化レベル 8 で 68 個，量子化レベル 16 で 1 個の有効な特徴量の組み合わせが，識別率 95%以上になった．③と④の境界線では，識別率 95%以上となる有効な特徴量の組み合わせはなかった．

さらに，7 章で有効だと判断した特徴量の識別率よりも識別率が高くなった特徴量の組み合わせを以下の表に示す．

表 8.8.1：7 章で評価した特徴量よりも識別率が高くなった 3 つの有効な特徴量の組み合わせ

評価した境界線：①と③の境界線　組み合わせ数：3

特徴量	特徴量の組み合わせ	識別率(%)	量子化レベル数	特徴量	特徴量の組み合わせ	識別率(%)	量子化レベル数	特徴量	特徴量の組み合わせ	識別率(%)	量子化レベル数
ASCII文字コード「0」(TPR：97.5%)	ASCII文字コード「NL*」 ASCII文字コード「0」 ASCII文字コード「p」	97.6	2	ASCII文字コード「o」(TPR：97 .0%)	HTTPリクエストの長さ ASCII文字コード「2」 ASCII文字コード「o」	97.5	2	ASCII文字コード「o」(TPR：97 .0%)	ASCII文字コード「C」 ASCII文字コード「e」 ASCII文字コード「o」	97.4	2
	ASCII文字コード「0」 ASCII文字コード「2」 ASCII文字コード「p」	97.8	2		ASCII文字コード「NL*」 ASCII文字コード「o」 ASCII文字コード「p」	97.6	2		ASCII文字コード「C」 ASCII文字コード「f」 ASCII文字コード「o」	97.5	2
	ASCII文字コード「0」 ASCII文字コード「C」 ASCII文字コード「p」	97.9	2		ASCII文字コード「0」 ASCII文字コード「C」 ASCII文字コード「o」	97.2	2		ASCII文字コード「C」 ASCII文字コード「o」 ASCII文字コード「p」	97.7	2
	ASCII文字コード「0」 ASCII文字コード「d」 ASCII文字コード「p」	98.1	2		ASCII文字コード「0」 ASCII文字コード「S」 ASCII文字コード「o」	97.3	2		ASCII文字コード「S」 ASCII文字コード「d」 ASCII文字コード「o」	97.5	2
	ASCII文字コード「0」 ASCII文字コード「e」 ASCII文字コード「p」	97.8	2		ASCII文字コード「0」 ASCII文字コード「e」 ASCII文字コード「o」	97.3	2		ASCII文字コード「S」 ASCII文字コード「e」 ASCII文字コード「o」	97.2	2
	ASCII文字コード「CR」 ASCII文字コード「0」 ASCII文字コード「5」	98.5	16		ASCII文字コード「1」 ASCII文字コード「C」 ASCII文字コード「o」	97.4	2		ASCII文字コード「S」 ASCII文字コード「f」 ASCII文字コード「o」	97.6	2
	ASCII文字コード「CR」 ASCII文字コード「0」 ASCII文字コード「C」	98.7	16		ASCII文字コード「1」 ASCII文字コード「S」 ASCII文字コード「o」	97.4	2		ASCII文字コード「S」 ASCII文字コード「o」 ASCII文字コード「p」	97.2	2
	ASCII文字コード「CR」 ASCII文字コード「0」 ASCII文字コード「i」	98.8	16		ASCII文字コード「1」 ASCII文字コード「d」 ASCII文字コード「o」	97.3	2		ASCII文字コード「S」 ASCII文字コード「o」 ASCII文字コード「r」	97.1	2
	ASCII文字コード「0」 ASCII文字コード「2」 ASCII文字コード「5」	98.4	16		ASCII文字コード「1」 ASCII文字コード「e」 ASCII文字コード「o」	97.5	2		ASCII文字コード「d」 ASCII文字コード「e」 ASCII文字コード「o」	97.2	2
	ASCII文字コード「0」 ASCII文字コード「2」 ASCII文字コード「C」	98.7	16		ASCII文字コード「2」 ASCII文字コード「C」 ASCII文字コード「o」	97.6	2		ASCII文字コード「d」 ASCII文字コード「f」 ASCII文字コード「o」	97.3	2
	ASCII文字コード「0」 ASCII文字コード「2」 ASCII文字コード「i」	98.9	16		ASCII文字コード「2」 ASCII文字コード「S」 ASCII文字コード「o」	97.6	2		ASCII文字コード「d」 ASCII文字コード「i」 ASCII文字コード「o」	97.7	2
ASCII文字コード「S」(TPR：98.0%)	HTTPリクエストの長さ ASCII文字コード「S」 ASCII文字コード「p」	98.5	2		ASCII文字コード「2」 ASCII文字コード「d」 ASCII文字コード「o」	97.4	2		ASCII文字コード「d」 ASCII文字コード「o」 ASCII文字コード「p」	97.9	2
	ASCII文字コード「S」 ASCII文字コード「d」 ASCII文字コード「p」	98.2	2		ASCII文字コード「2」 ASCII文字コード「e」 ASCII文字コード「o」	97.4	2		ASCII文字コード「e」 ASCII文字コード「f」 ASCII文字コード「o」	97.3	2
	ASCII文字コード「S」 ASCII文字コード「e」 ASCII文字コード「p」	98.2	2		ASCII文字コード「2」 ASCII文字コード「o」 ASCII文字コード「p」	98	2		ASCII文字コード「e」 ASCII文字コード「o」 ASCII文字コード「p」	97.8	2
	ASCII文字コード「1」 ASCII文字コード「S」 ASCII文字コード「i」	98.8	16		ASCII文字コード「2」 ASCII文字コード「o」 ASCII文字コード「x」	97.1	2		ASCII文字コード「e」 ASCII文字コード「o」 ASCII文字コード「r」	97.2	2
ASCII文字コード「f」(TPR：97.5%)	HTTPリクエストの長さ ASCII文字コード「f」 ASCII文字コード「p」	98.7	2		ASCII文字コード「C」 ASCII文字コード「S」 ASCII文字コード「o」	97.6	2		ASCII文字コード「e」 ASCII文字コード「o」 ASCII文字コード「x」	97.2	2
	ASCII文字コード「NL*」 ASCII文字コード「f」 ASCII文字コード「p」	97.7	2		ASCII文字コード「C」 ASCII文字コード「d」 ASCII文字コード「o」	97.1	2		HTTPリクエストの長さ ASCII文字コード「o」 ASCII文字コード「p」	97.2	4
	ASCII文字コード「S」 ASCII文字コード「f」 ASCII文字コード「o」	97.6	2								
	ASCII文字コード「S」 ASCII文字コード「f」 ASCII文字コード「p」	97.7	2								
ASCII文字コード「i」(TPR：98.5%)(TNR：83.0%)	HTTPリクエストの長さ ASCII文字コード「i」 ASCII文字コード「p」	98.6	2								
	ASCII文字コード「1」 ASCII文字コード「S」 ASCII文字コード「i」	98.8	2								
	ASCII文字コード「1」 ASCII文字コード「i」 ASCII文字コード「x」	99.2	16								

表 8.8.2：7 章で評価した特徴量よりも識別率が高くなった 3 つの有効な特徴量の組み合わせ

評価した境界線:①と②の境界線　組み合わせ数:3

特徴量	特徴量の 組み合わせ	識別率 (%)	量子化 レベル数
ASCII文字コード「p」 （TPR：99.0%）	ASCII文字コード「CR」 ASCII文字コード「d」 ASCII文字コード「p」	99.4	16
	ASCII文字コード「CR」 ASCII文字コード「f」 ASCII文字コード「p」	99.3	16
	ASCII文字コード「2」 ASCII文字コード「A」 ASCII文字コード「p」	99.1	16
	ASCII文字コード「2」 ASCII文字コード「e」 ASCII文字コード「p」	99.1	16
	ASCII文字コード「2」 ASCII文字コード「f」 ASCII文字コード「p」	99.1	16
	ASCII文字コード「2」 ASCII文字コード「p」 ASCII文字コード「t」	99.1	16
	ASCII文字コード「A」 ASCII文字コード「C」 ASCII文字コード「p」	99.1	16
	ASCII文字コード「A」 ASCII文字コード「d」 ASCII文字コード「p」	99.5	16
	ASCII文字コード「A」 ASCII文字コード「f」 ASCII文字コード「p」	99.7	16
	ASCII文字コード「C」 ASCII文字コード「d」 ASCII文字コード「p」	99.4	16
	ASCII文字コード「d」 ASCII文字コード「e」 ASCII文字コード「p」	99.5	16
	ASCII文字コード「d」 ASCII文字コード「f」 ASCII文字コード「p」	99.1	16
	ASCII文字コード「d」 ASCII文字コード「i」 ASCII文字コード「p」	99.5	16
	ASCII文字コード「d」 ASCII文字コード「p」 ASCII文字コード「t」	99.3	16
	ASCII文字コード「d」 ASCII文字コード「p」 ASCII文字コード「x」	99.3	16
	ASCII文字コード「e」 ASCII文字コード「f」 ASCII文字コード「p」	99.3	16
	ASCII文字コード「f」 ASCII文字コード「i」 ASCII文字コード「p」	99.2	16
	ASCII文字コード「f」 ASCII文字コード「o」 ASCII文字コード「p」	99.1	16
	ASCII文字コード「f」 ASCII文字コード「p」 ASCII文字コード「x」	99.2	16

表 8.8.3：7 章で評価した特徴量よりも識別率が高くなった 3 つの有効な特徴量の組み合わせ

評価した境界線：①と②の境界線　組み合わせ数：3

特徴量	特徴量の組み合わせ	識別率(%)	量子化レベル数	特徴量	特徴量の組み合わせ	識別率(%)	量子化レベル数
ASCII文字コード「0」(TPR：97.5%)	ASCII文字コード「0」 ASCII文字コード「2」 ASCII文字コード「5」	98.4	16	ASCII文字コード「e」(TPR：99.0%)	ASCII文字コード「e」 ASCII文字コード「i」 ASCII文字コード「o」	99.5	2
	ASCII文字コード「0」 ASCII文字コード「2」 ASCII文字コード「C」	98.7	16		ASCII文字コード「e」 ASCII文字コード「i」 ASCII文字コード「r」	99.4	2
	ASCII文字コード「0」 ASCII文字コード「2」 ASCII文字コード「i」	98.9	16		ASCII文字コード「e」 ASCII文字コード「i」 ASCII文字コード「t」	99.5	2
	ASCII文字コード「0」 ASCII文字コード「5」 ASCII文字コード「e」	98.5	16		ASCII文字コード「e」 ASCII文字コード「o」 ASCII文字コード「r」	99.5	2
	ASCII文字コード「0」 ASCII文字コード「5」 ASCII文字コード「i」	99	16		ASCII文字コード「e」 ASCII文字コード「o」 ASCII文字コード「t」	99.3	2
	ASCII文字コード「0」 ASCII文字コード「5」 ASCII文字コード「x」	98.4	16		ASCII文字コード「CR」 ASCII文字コード「2」 ASCII文字コード「e」	99.1	8
	ASCII文字コード「0」 ASCII文字コード「C」 ASCII文字コード「e」	98.2	16		ASCII文字コード「CR」 ASCII文字コード「e」 ASCII文字コード「f」	99.2	8
	ASCII文字コード「0」 ASCII文字コード「C」 ASCII文字コード「i」	98	16		ASCII文字コード「CR」 ASCII文字コード「e」 ASCII文字コード「x」	99.1	8
	ASCII文字コード「0」 ASCII文字コード「C」 ASCII文字コード「x」	98.5	16		ASCII文字コード「2」 ASCII文字コード「5」 ASCII文字コード「e」	99.4	8
	ASCII文字コード「0」 ASCII文字コード「i」 ASCII文字コード「x」	98.9	16		ASCII文字コード「2」 ASCII文字コード「d」 ASCII文字コード「e」	99.5	8
ASCII文字コード「e」(TPR：99.0%)	ASCII文字コード「NL*」 ASCII文字コード「d」 ASCII文字コード「e」	99.5	2		ASCII文字コード「2」 ASCII文字コード「e」 ASCII文字コード「i」	99.1	8
	ASCII文字コード「NL*」 ASCII文字コード「e」 ASCII文字コード「o」	99.1	2		ASCII文字コード「2」 ASCII文字コード「e」 ASCII文字コード「p」	99.1	8
	ASCII文字コード「NL*」 ASCII文字コード「e」 ASCII文字コード「r」	99.5	2		ASCII文字コード「2」 ASCII文字コード「e」 ASCII文字コード「t」	99.1	8
	ASCII文字コード「NL*」 ASCII文字コード「e」 ASCII文字コード「t」	99.5	2		ASCII文字コード「5」 ASCII文字コード「A」 ASCII文字コード「e」	99.7	8
	ASCII文字コード「CR」 ASCII文字コード「d」 ASCII文字コード「e」	99.5	2		ASCII文字コード「5」 ASCII文字コード「C」 ASCII文字コード「e」	99.4	8
		99.5	4			99.4	16
		99.5	8		ASCII文字コード「5」 ASCII文字コード「e」 ASCII文字コード「f」	99.3	8
	ASCII文字コード「CR」 ASCII文字コード「e」 ASCII文字コード「i」	99.2	2		ASCII文字コード「5」 ASCII文字コード「e」 ASCII文字コード「i」	99.1	8
	ASCII文字コード「CR」 ASCII文字コード「e」 ASCII文字コード「o」	99.2	2		ASCII文字コード「A」 ASCII文字コード「C」 ASCII文字コード「e」	99.4	8
	ASCII文字コード「CR」 ASCII文字コード「e」 ASCII文字コード「r」	99.9	2		ASCII文字コード「A」 ASCII文字コード「d」 ASCII文字コード「e」	99.1	8
	ASCII文字コード「CR」 ASCII文字コード「e」 ASCII文字コード「t」	99.2	2		ASCII文字コード「A」 ASCII文字コード「e」 ASCII文字コード「f」	99.4	8
		99.1	8		ASCII文字コード「A」 ASCII文字コード「e」 ASCII文字コード「f」	99.7	8
	ASCII文字コード「d」 ASCII文字コード「e」 ASCII文字コード「i」	99.5	2		ASCII文字コード「C」 ASCII文字コード「e」 ASCII文字コード「f」	99.1	8
		99.2	8		ASCII文字コード「d」 ASCII文字コード「e」 ASCII文字コード「f」	99.1	8
	ASCII文字コード「d」 ASCII文字コード「e」 ASCII文字コード「o」	99.5	2		ASCII文字コード「d」 ASCII文字コード「e」 ASCII文字コード「x」	99.1	8
	ASCII文字コード「d」 ASCII文字コード「e」 ASCII文字コード「r」	99.5	2		ASCII文字コード「e」 ASCII文字コード「f」 ASCII文字コード「i」	99.3	8
	ASCII文字コード「d」 ASCII文字コード「e」 ASCII文字コード「t」	99.5	2		ASCII文字コード「e」 ASCII文字コード「f」 ASCII文字コード「o」	99.2	8
		99.4	4		ASCII文字コード「e」 ASCII文字コード「f」 ASCII文字コード「p」	99.3	8
		99.3	8		ASCII文字コード「e」 ASCII文字コード「f」 ASCII文字コード「t」	99.1	8
					ASCII文字コード「e」 ASCII文字コード「f」 ASCII文字コード「x」	99.1	8
					ASCII文字コード「e」 ASCII文字コード「t」 ASCII文字コード「x」	99.1	8

表 8.8.4：7 章で評価した特徴量よりも識別率が高くなった 3 つの有効な特徴量の組み合わせ

評価した境界線：①と②の境界線　組み合わせ数：3

特徴量	特徴量の組み合わせ	識別率(%)	量子化レベル数	特徴量	特徴量の組み合わせ	識別率(%)	量子化レベル数
ASCII文字コード「f」(TPR：97.5%)	ASCII文字コード「CR」 ASCII文字コード「2」 ASCII文字コード「f」	99.1	8	ASCII文字コード「f」(TPR：97.5%)	ASCII文字コード「A」 ASCII文字コード「f」 ASCII文字コード「p」	99.7	8
	ASCII文字コード「CR」 ASCII文字コード「5」 ASCII文字コード「f」	99.1	8		ASCII文字コード「A」 ASCII文字コード「f」 ASCII文字コード「t」	99.5	8
	ASCII文字コード「CR」 ASCII文字コード「A」 ASCII文字コード「f」	99.7	8		ASCII文字コード「A」 ASCII文字コード「f」 ASCII文字コード「x」	99.5	8
	ASCII文字コード「CR」 ASCII文字コード「C」 ASCII文字コード「f」	99.1	8		ASCII文字コード「C」 ASCII文字コード「d」 ASCII文字コード「f」	99	8
	ASCII文字コード「CR」 ASCII文字コード「d」 ASCII文字コード「f」	99	8		ASCII文字コード「C」 ASCII文字コード「e」 ASCII文字コード「f」	99.1	8
	ASCII文字コード「CR」 ASCII文字コード「e」 ASCII文字コード「f」	99.2	8		ASCII文字コード「C」 ASCII文字コード「f」 ASCII文字コード「i」	99.2	8
	ASCII文字コード「CR」 ASCII文字コード「f」 ASCII文字コード「i」	99.3	8		ASCII文字コード「C」 ASCII文字コード「f」 ASCII文字コード「o」	99.1	8
	ASCII文字コード「CR」 ASCII文字コード「f」 ASCII文字コード「o」	99.1	8		ASCII文字コード「C」 ASCII文字コード「f」 ASCII文字コード「p」	99	8
	ASCII文字コード「CR」 ASCII文字コード「f」 ASCII文字コード「p」	99.3	8		ASCII文字コード「C」 ASCII文字コード「p」 ASCII文字コード「t」	99	8
	ASCII文字コード「CR」 ASCII文字コード「f」 ASCII文字コード「t」	99.1	8		ASCII文字コード「C」 ASCII文字コード「p」 ASCII文字コード「x」	99	8
	ASCII文字コード「CR」 ASCII文字コード「f」 ASCII文字コード「x」	99.1	8		ASCII文字コード「d」 ASCII文字コード「e」 ASCII文字コード「f」	99.1	8
	ASCII文字コード「2」 ASCII文字コード「5」 ASCII文字コード「f」	99	8		ASCII文字コード「d」 ASCII文字コード「f」 ASCII文字コード「o」	99.2	8
	ASCII文字コード「2」 ASCII文字コード「A」 ASCII文字コード「f」	99.7	8		ASCII文字コード「d」 ASCII文字コード「f」 ASCII文字コード「o」	98.9	8
	ASCII文字コード「2」 ASCII文字コード「C」 ASCII文字コード「f」	98.9	8		ASCII文字コード「d」 ASCII文字コード「f」 ASCII文字コード「p」	99.1	8
	ASCII文字コード「2」 ASCII文字コード「d」 ASCII文字コード「f」	99.1	8		ASCII文字コード「d」 ASCII文字コード「f」 ASCII文字コード「t」	99	8
	ASCII文字コード「2」 ASCII文字コード「e」 ASCII文字コード「f」	99	8		ASCII文字コード「d」 ASCII文字コード「f」 ASCII文字コード「x」	98.9	8
	ASCII文字コード「2」 ASCII文字コード「f」 ASCII文字コード「i」	99.1	8		ASCII文字コード「e」 ASCII文字コード「f」 ASCII文字コード「i」	99.3	8
	ASCII文字コード「2」 ASCII文字コード「f」 ASCII文字コード「o」	99.3	8		ASCII文字コード「e」 ASCII文字コード「f」 ASCII文字コード「o」	99.2	8
	ASCII文字コード「2」 ASCII文字コード「d」 ASCII文字コード「t」	99.2	8		ASCII文字コード「e」 ASCII文字コード「f」 ASCII文字コード「p」	99.3	8
	ASCII文字コード「2」 ASCII文字コード「d」 ASCII文字コード「x」	99	8		ASCII文字コード「e」 ASCII文字コード「f」 ASCII文字コード「t」	99.1	8
	ASCII文字コード「5」 ASCII文字コード「C」 ASCII文字コード「f」	99	8		ASCII文字コード「e」 ASCII文字コード「f」 ASCII文字コード「x」	99.1	8
	ASCII文字コード「5」 ASCII文字コード「d」 ASCII文字コード「f」	99.5	8		ASCII文字コード「f」 ASCII文字コード「i」 ASCII文字コード「o」	99.2	8
	ASCII文字コード「5」 ASCII文字コード「e」 ASCII文字コード「f」	99.3	8		ASCII文字コード「f」 ASCII文字コード「i」 ASCII文字コード「p」	99.2	8
	ASCII文字コード「5」 ASCII文字コード「f」 ASCII文字コード「i」	99	8		ASCII文字コード「f」 ASCII文字コード「i」 ASCII文字コード「t」	99.2	8
	ASCII文字コード「5」 ASCII文字コード「f」 ASCII文字コード「o」	99.3	8		ASCII文字コード「f」 ASCII文字コード「i」 ASCII文字コード「x」	99.2	8
	ASCII文字コード「5」 ASCII文字コード「f」 ASCII文字コード「p」	99	8		ASCII文字コード「f」 ASCII文字コード「o」 ASCII文字コード「p」	99.1	8
	ASCII文字コード「5」 ASCII文字コード「f」 ASCII文字コード「t」	99.2	8		ASCII文字コード「f」 ASCII文字コード「o」 ASCII文字コード「t」	99.1	8
	ASCII文字コード「5」 ASCII文字コード「f」 ASCII文字コード「x」	99	8		ASCII文字コード「f」 ASCII文字コード「o」 ASCII文字コード「x」	99.1	8
	ASCII文字コード「A」 ASCII文字コード「C」 ASCII文字コード「f」	99.1	8		ASCII文字コード「f」 ASCII文字コード「p」 ASCII文字コード「t」	99	8
	ASCII文字コード「A」 ASCII文字コード「d」 ASCII文字コード「f」	99.4	8		ASCII文字コード「f」 ASCII文字コード「p」 ASCII文字コード「x」	99.2	8
	ASCII文字コード「A」 ASCII文字コード「e」 ASCII文字コード「f」	99.4	8		ASCII文字コード「f」 ASCII文字コード「t」 ASCII文字コード「x」	99	8
	ASCII文字コード「A」 ASCII文字コード「f」 ASCII文字コード「o」	99.1	8				

表 8.8.5：7 章で評価した特徴量よりも識別率が高くなった 3 つの有効な特徴量の組み合わせ

評価した境界線:①と②の境界線　組み合わせ数:3

特徴量	特徴量の組み合わせ	識別率(%)	量子化レベル数	特徴量	特徴量の組み合わせ	識別率(%)	量子化レベル数
ASCII文字コード「i」(TPR : 98.5%)(TNR : 83.0%)	ASCII文字コード「NL*」ASCII文字コード「d」ASCII文字コード「i」	99.2	2	ASCII文字コード「i」(TPR : 98.5%)(TNR : 83.0%)	ASCII文字コード「2」ASCII文字コード「i」ASCII文字コード「x」	98.8	8
	ASCII文字コード「NL*」ASCII文字コード「e」ASCII文字コード「i」	99.1	2		ASCII文字コード「5」ASCII文字コード「A」ASCII文字コード「i」	99.7	8
	ASCII文字コード「NL*」ASCII文字コード「i」ASCII文字コード「o」	99.5	2		ASCII文字コード「5」ASCII文字コード「C」ASCII文字コード「i」	99	8
	ASCII文字コード「NL*」ASCII文字コード「i」ASCII文字コード「t」	99.5	2			98.9	16
	ASCII文字コード「CR」ASCII文字コード「d」ASCII文字コード「i」	99.4	2		ASCII文字コード「5」ASCII文字コード「e」ASCII文字コード「i」	99.1	8
		99.2	8		ASCII文字コード「5」ASCII文字コード「f」ASCII文字コード「i」	99	8
	ASCII文字コード「CR」ASCII文字コード「e」ASCII文字コード「i」	99.5	2		ASCII文字コード「5」ASCII文字コード「i」ASCII文字コード「o」	98.8	8
	ASCII文字コード「CR」ASCII文字コード「i」ASCII文字コード「o」	99.4	2		ASCII文字コード「5」ASCII文字コード「i」ASCII文字コード「t」	99	8
	ASCII文字コード「CR」ASCII文字コード「i」ASCII文字コード「t」	99.5	2		ASCII文字コード「A」ASCII文字コード「C」ASCII文字コード「i」	99.7	8
	ASCII文字コード「d」ASCII文字コード「i」ASCII文字コード「o」	99.5	2		ASCII文字コード「A」ASCII文字コード「d」ASCII文字コード「i」	99.5	8
		99.1	8		ASCII文字コード「A」ASCII文字コード「e」ASCII文字コード「i」	99.7	8
	ASCII文字コード「d」ASCII文字コード「i」ASCII文字コード「t」	99.5	2		ASCII文字コード「A」ASCII文字コード「i」ASCII文字コード「t」	98.8	8
		99.3	8		ASCII文字コード「C」ASCII文字コード「d」ASCII文字コード「i」	99.2	8
	ASCII文字コード「e」ASCII文字コード「i」ASCII文字コード「o」	99.5	2		ASCII文字コード「C」ASCII文字コード「e」ASCII文字コード「i」	98.6	8
		99.5	8			98.6	16
	ASCII文字コード「e」ASCII文字コード「i」ASCII文字コード「r」	99.4	2		ASCII文字コード「C」ASCII文字コード「f」ASCII文字コード「i」	99.2	8
	ASCII文字コード「e」ASCII文字コード「i」ASCII文字コード「t」	99.5	2		ASCII文字コード「C」ASCII文字コード「i」ASCII文字コード「p」	99	8
	ASCII文字コード「i」ASCII文字コード「o」ASCII文字コード「t」	99.5	2		ASCII文字コード「C」ASCII文字コード「i」ASCII文字コード「x」	99	8
	ASCII文字コード「o」ASCII文字コード「r」ASCII文字コード「t」	99.5	2		ASCII文字コード「d」ASCII文字コード「f」ASCII文字コード「i」	99.2	8
	ASCII文字コード「CR」ASCII文字コード「2」ASCII文字コード「i」	99	8		ASCII文字コード「d」ASCII文字コード「i」ASCII文字コード「p」	99.5	8
		98.9	16		ASCII文字コード「d」ASCII文字コード「i」ASCII文字コード「x」	99.1	8
	ASCII文字コード「CR」ASCII文字コード「5」ASCII文字コード「i」	99	8		ASCII文字コード「e」ASCII文字コード「f」ASCII文字コード「i」	99.3	8
		99	16		ASCII文字コード「e」ASCII文字コード「i」ASCII文字コード「p」	99	8
	ASCII文字コード「CR」ASCII文字コード「A」ASCII文字コード「i」	98.6	8		ASCII文字コード「f」ASCII文字コード「i」ASCII文字コード「o」	99.2	8
	ASCII文字コード「CR」ASCII文字コード「C」ASCII文字コード「i」	98.8	8		ASCII文字コード「f」ASCII文字コード「i」ASCII文字コード「p」	99.2	8
	ASCII文字コード「CR」ASCII文字コード「f」ASCII文字コード「i」	99.3	8		ASCII文字コード「f」ASCII文字コード「i」ASCII文字コード「t」	99.2	8
	ASCII文字コード「CR」ASCII文字コード「i」ASCII文字コード「p」	99	8		ASCII文字コード「f」ASCII文字コード「i」ASCII文字コード「x」	99.2	8
	ASCII文字コード「CR」ASCII文字コード「i」ASCII文字コード「x」	99	8		ASCII文字コード「i」ASCII文字コード「o」ASCII文字コード「p」	98.8	8
		98.9	16		ASCII文字コード「i」ASCII文字コード「p」ASCII文字コード「t」	98.6	8
	ASCII文字コード「2」ASCII文字コード「5」ASCII文字コード「i」	99.2	8		ASCII文字コード「i」ASCII文字コード「p」ASCII文字コード「x」	99	8
		99	16		ASCII文字コード「CR」ASCII文字コード「0」ASCII文字コード「i」	98.8	16
	ASCII文字コード「2」ASCII文字コード「A」ASCII文字コード「i」	99.7	8		ASCII文字コード「CR」ASCII文字コード「C」ASCII文字コード「i」	98.8	16
	ASCII文字コード「2」ASCII文字コード「C」ASCII文字コード「i」	99	8		ASCII文字コード「0」ASCII文字コード「2」ASCII文字コード「i」	98.9	16
		98.7	16		ASCII文字コード「0」ASCII文字コード「5」ASCII文字コード「i」	99	16
	ASCII文字コード「2」ASCII文字コード「d」ASCII文字コード「i」	99	8		ASCII文字コード「0」ASCII文字コード「i」ASCII文字コード「x」	98.9	16
	ASCII文字コード「2」ASCII文字コード「e」ASCII文字コード「i」	99.1	8		ASCII文字コード「5」ASCII文字コード「d」ASCII文字コード「i」	99.1	16
	ASCII文字コード「2」ASCII文字コード「f」ASCII文字コード「i」	99.1	8		ASCII文字コード「5」ASCII文字コード「i」ASCII文字コード「x」	99	16
	ASCII文字コード「2」ASCII文字コード「i」ASCII文字コード「t」	99	8				

表 8.8.6：7 章で評価した特徴量よりも識別率が高くなった 3 つの有効な特徴量の組み合わせ

評価した境界線：①と②の境界線　組み合わせ数：3

特徴量	特徴量の組み合わせ	識別率(%)	量子化レベル数	特徴量	特徴量の組み合わせ	識別率(%)	量子化レベル数
ASCII文字コード「o」(TPR：97.0%)	ASCII文字コード「NL*」 ASCII文字コード「d」 ASCII文字コード「o」	99.3	2	ASCII文字コード「o」(TPR：97.0%)	ASCII文字コード「2」 ASCII文字コード「d」 ASCII文字コード「o」	99.1	8
		99	4		ASCII文字コード「2」 ASCII文字コード「f」 ASCII文字コード「o」	99.3	8
	ASCII文字コード「NL*」 ASCII文字コード「e」 ASCII文字コード「o」	99.5	2		ASCII文字コード「2」 ASCII文字コード「o」 ASCII文字コード「p」	98.7	8
	ASCII文字コード「NL*」 ASCII文字コード「i」 ASCII文字コード「o」	99.5	2		ASCII文字コード「2」 ASCII文字コード「o」 ASCII文字コード「t」	99.1	8
	ASCII文字コード「NL*」 ASCII文字コード「o」 ASCII文字コード「r」	99.5	2		ASCII文字コード「2」 ASCII文字コード「o」 ASCII文字コード「x」	98.9	8
	ASCII文字コード「NL*」 ASCII文字コード「o」 ASCII文字コード「t」	99.5	2		ASCII文字コード「5」 ASCII文字コード「i」 ASCII文字コード「o」	98.4	8
	ASCII文字コード「CR」 ASCII文字コード「d」 ASCII文字コード「o」	99.5	2		ASCII文字コード「5」 ASCII文字コード「C」 ASCII文字コード「o」	98.9	8
		98.8	4		ASCII文字コード「5」 ASCII文字コード「d」 ASCII文字コード「o」	99.5	8
		99	8		ASCII文字コード「5」 ASCII文字コード「e」 ASCII文字コード「o」	98.8	8
	ASCII文字コード「CR」 ASCII文字コード「e」 ASCII文字コード「o」	99.2	2		ASCII文字コード「5」 ASCII文字コード「f」 ASCII文字コード「o」	99.3	8
		97.9	8		ASCII文字コード「5」 ASCII文字コード「i」 ASCII文字コード「o」	98.8	8
	ASCII文字コード「CR」 ASCII文字コード「i」 ASCII文字コード「o」	99.4	2		ASCII文字コード「5」 ASCII文字コード「o」 ASCII文字コード「p」	98.5	8
	ASCII文字コード「CR」 ASCII文字コード「o」 ASCII文字コード「r」	99.6	2		ASCII文字コード「5」 ASCII文字コード「o」 ASCII文字コード「t」	98.7	8
		98.9	4		ASCII文字コード「A」 ASCII文字コード「d」 ASCII文字コード「o」	99.6	8
	ASCII文字コード「CR」 ASCII文字コード「o」 ASCII文字コード「t」	99.3	2		ASCII文字コード「A」 ASCII文字コード「f」 ASCII文字コード「o」	99.1	8
		98.3	8		ASCII文字コード「A」 ASCII文字コード「o」 ASCII文字コード「r」	98	8
	ASCII文字コード「d」 ASCII文字コード「e」 ASCII文字コード「o」	99.5	2		ASCII文字コード「A」 ASCII文字コード「o」 ASCII文字コード「t」	97.8	8
		99.2	4		ASCII文字コード「C」 ASCII文字コード「d」 ASCII文字コード「o」	99	8
		99	8		ASCII文字コード「C」 ASCII文字コード「e」 ASCII文字コード「o」	98.4	8
	ASCII文字コード「d」 ASCII文字コード「i」 ASCII文字コード「o」	99.5	2		ASCII文字コード「C」 ASCII文字コード「f」 ASCII文字コード「o」	99.1	8
		99.1	8		ASCII文字コード「C」 ASCII文字コード「i」 ASCII文字コード「o」	98.4	8
	ASCII文字コード「d」 ASCII文字コード「o」 ASCII文字コード「r」	99.3	2		ASCII文字コード「C」 ASCII文字コード「o」 ASCII文字コード「p」	98.4	8
		99.2	4		ASCII文字コード「C」 ASCII文字コード「o」 ASCII文字コード「t」	98.5	8
	ASCII文字コード「d」 ASCII文字コード「o」 ASCII文字コード「t」	99	2		ASCII文字コード「C」 ASCII文字コード「o」 ASCII文字コード「x」	98.3	8
		99.4	4		ASCII文字コード「d」 ASCII文字コード「f」 ASCII文字コード「o」	98.9	8
		99.3	8		ASCII文字コード「d」 ASCII文字コード「o」 ASCII文字コード「p」	99	8
	ASCII文字コード「e」 ASCII文字コード「i」 ASCII文字コード「o」	99.5	2		ASCII文字コード「d」 ASCII文字コード「o」 ASCII文字コード「x」	99	8
		98.8	8		ASCII文字コード「e」 ASCII文字コード「f」 ASCII文字コード「o」	99.2	8
	ASCII文字コード「e」 ASCII文字コード「o」 ASCII文字コード「t」	99.3	2		ASCII文字コード「e」 ASCII文字コード「o」 ASCII文字コード「p」	98.6	8
		99.4	4		ASCII文字コード「e」 ASCII文字コード「o」 ASCII文字コード「x」	99	8
		98.6	8		ASCII文字コード「f」 ASCII文字コード「i」 ASCII文字コード「o」	99.2	8
	ASCII文字コード「o」 ASCII文字コード「r」 ASCII文字コード「t」	99.4	2		ASCII文字コード「f」 ASCII文字コード「o」 ASCII文字コード「p」	99.1	8
	ASCII文字コード「CR」 ASCII文字コード「2」 ASCII文字コード「o」	99.1	8		ASCII文字コード「f」 ASCII文字コード「o」 ASCII文字コード「t」	99.1	8
	ASCII文字コード「CR」 ASCII文字コード「A」 ASCII文字コード「o」	99.3	8		ASCII文字コード「f」 ASCII文字コード「o」 ASCII文字コード「x」	99.1	8
	ASCII文字コード「CR」 ASCII文字コード「C」 ASCII文字コード「o」	98.5	8		ASCII文字コード「i」 ASCII文字コード「o」 ASCII文字コード「p」	98.8	8
	ASCII文字コード「CR」 ASCII文字コード「f」 ASCII文字コード「o」	99.1	8		ASCII文字コード「o」 ASCII文字コード「p」 ASCII文字コード「t」	98	8
	ASCII文字コード「CR」 ASCII文字コード「o」 ASCII文字コード「p」	98.9	8		ASCII文字コード「o」 ASCII文字コード「p」 ASCII文字コード「x」	98.3	8
	ASCII文字コード「CR」 ASCII文字コード「o」 ASCII文字コード「x」	99	8		ASCII文字コード「o」 ASCII文字コード「t」 ASCII文字コード「x」	98.5	8
	ASCII文字コード「2」 ASCII文字コード「C」 ASCII文字コード「o」	99	8				

表 8.8.1，8.8.2，8.8.3，8.8.4，8.8.5，8.8.6 より，以下の有効な特徴量の組み合わせが， 7 章で有効だと判断した特徴量の識別率よりも，識別率が高くなった．

- ①と③の境界線：
  ASCII 文字コード「0」で 11 つ，ASCII 文字コード「S」で 4 つ，ASCII 文字コード「f」で 4 つ，ASCII 文字コード「i」で 3 つ，ASCII 文字コード「o」で 34 つ
- ①と②の境界線：
  ASCII 文字コード「S」で 13 つ，ASCII 文字コード「e」で 43 つ，ASCII 文字コード「f」で 63 つ，ASCII 文字コード「i」で 65 つ，ASCII 文字コード「o」で 62 つ， ASCII 文字コード「p」で 19 つ

さらにその中でも，2 つの特徴量の識別率が共に向上した組み合わせが以下の通りになった．

- ①と②の境界線：7 つ
  - ASCII 文字コード「e」と ASCII 文字コード「i」と ASCII 文字コード「o」：
    （ASCII 文字コード「e」: 99.0% → 99.5%，ASCII 文字コード「i」: 98.5% → 99.5%，
    ASCII 文字コード「o」: 97.0% → 99.5%）
  - ASCII 文字コード「e」と ASCII 文字コード「f」と ASCII 文字コード「o」：
    （ASCII 文字コード「e」: 99.0% → 99.2%， ASCII 文字コード「f」: 97.5% → 99.2%
    ASCII 文字コード「o」: 97.0% → 99.2%）
  - ASCII 文字コード「e」と ASCII 文字コード「f」と ASCII 文字コード「o」：
    （ASCII 文字コード「e」: 99.0% → 99.2%， ASCII 文字コード「f」: 97.5% → 99.2%
    ASCII 文字コード「o」: 97.0% → 99.2%）
  - ASCII 文字コード「e」と ASCII 文字コード「f」と ASCII 文字コード「p」：
    （ASCII 文字コード「e」: 99.0% → 99.3%， ASCII 文字コード「f」: 97.5% → 99.3%
    ASCII 文字コード「p」: 99.0% → 99.2%）
  - ASCII 文字コード「f」と ASCII 文字コード「i」と ASCII 文字コード「o」：
    （ASCII 文字コード「f」: 97.5% → 99.2%，ASCII 文字コード「i」: 98.5% → 99.2%，
    ASCII 文字コード「o」: 97.0% → 99.2%）
  - ASCII 文字コード「f」と ASCII 文字コード「i」と ASCII 文字コード「p」：
    （ASCII 文字コード「f」: 97.5% → 99.2%，ASCII 文字コード「i」: 98.5% → 99.2%，
    ASCII 文字コード「o」: 97.0% → 99.2%）
  - ASCII 文字コード「f」と ASCII 文字コード「o」と ASCII 文字コード「p」：
    （ASCII 文字コード「f」: 97.5% → 99.1%，ASCII 文字コード「o」: 97.0% → 99.1%，
    ASCII 文字コード「p」: 99.0% → 99.1%）

8.3 考察

8.2 節より，7 章で有効だと判断した特徴量よりも識別率が高くなった特徴量の組み合わせについて，正常時通信と感染時通信の重なり具合を視覚的に確認し，考察した．中でも，識別率がより高くなった ASCII 文字コード「o」について着目する．具体的には，2 つの特徴量（ASCII 文字コード「0」，ASCII 文字コード「e」）を組み合わせることで，97.0%から 100%に向上した．

ASCII 文字コード「o」「e」に着目した時，感染時通信の ASCII 文字コード「o」「e」の出現頻度が 1 スロット数 10 未満で多く分布しているのに対し，正常時通信の ASCII 文字コード「o」「e」の出現頻度が 1 スロット 1000 以上出現することがある．そのため，ASCII 文字コード「o」「e」は感染コードブックが小さい値で，正常コードブックが大きい値で作成された．

これは 7 章で考察したことと同様に，正常時通信と感染時通信のペイロード情報の違いで特徴量の出現頻度が異なるからである．

この作成されたコードブックとテストデータとのスコア，モデルの結果から，モデルの学習結果の値と式(8.9)を用い，スコアの分布図および①と③の境界線（正常時通信と感染時通信を分類する境界線）を作成した（図 8.2）．

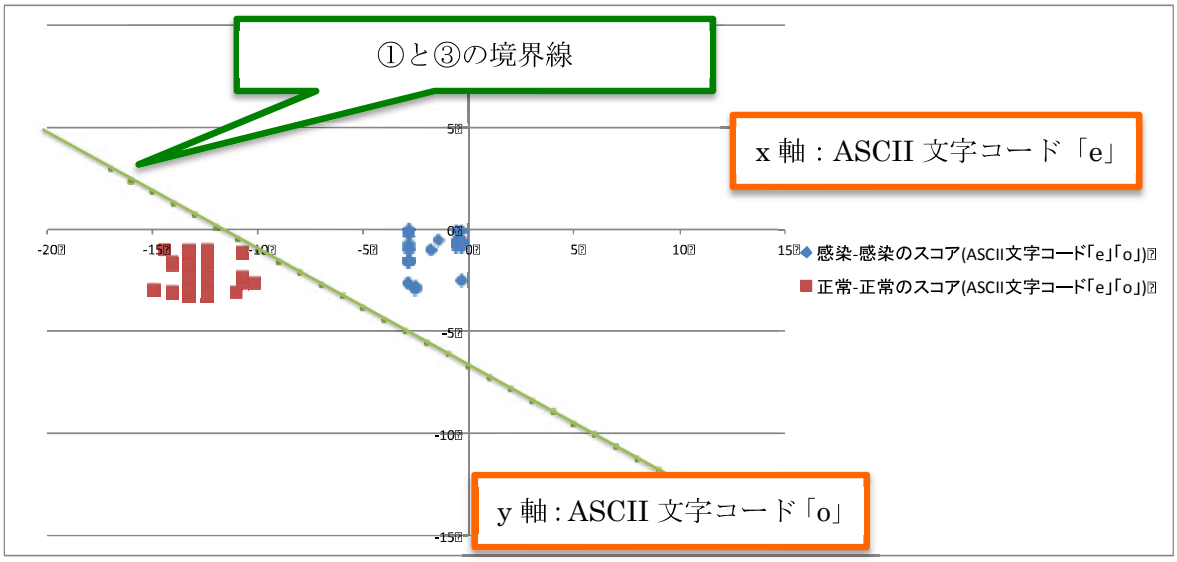

図 8.2：ロジスティック回帰モデルを用いた①と③の境界線

スコアの分布図から境界線によって，正常と感染を上手く分類できていることがわかる．これは，境界線を作成するとき，正常と感染のスコアを学習した重みづけの値（$\omega_1 = -0.084814$, $\omega_2 = -0.149675$）が正常と感染を上手く分類できる値になったからだと考えられる．

また 6 章より，ASCII 文字コード「e」は，トロイの木馬，ファイル感染型ウイルスを検知するのに有効な特徴量であり，また，ASCII 文字コード「o」，トロイの木馬を検知するのに有効な特徴量であったが，この 2 つの特徴量を組み合わせることによって，ワーム，トロイの木馬，ファイル感染型ウイルスの 3 種類のマルウェアを検知できるようになった．これも，境界線を作成するとき，正常と感染のスコアを学習した重みづけの値が正常と感染を上手く分類できる値になったからだと考えられる．

そのため，この 2 つの特徴量のスコアを組み合わせたものを学習した学習モデルの精度が上がり，テストモデルを正常と感染に上手く分類でき，識別率が向上したと考えられる．

第 9 章　結論

昨今，未知のマルウェアの増加，またそれによる感染被害が問題となっている．これに対し，シグネチャ型などの手法を用いてマルウェアの検知，駆除を行うマルウェア対策ソフトがセキュリティベンダによって開発されている．しかしこれらのマルウェア対策ソフトの検知手法は，マルウェア毎に特徴を示すシグネチャを用意しなければならず，短期間で大量に出現する未知のマルウェアの検知に対応できない．そのため，感染してしまうことを前提として，感染後に早期に検知する技術（感染検知）が必要とされている．

また，未知マルウェアを検知するための方法として，マルウェアに感染した PC はマルウェア感染時に特有な通信（トラヒックデータ）を行うことが考えられる．よって，PC の通信に利用されるトラヒックデータに着目して未知のマルウェアを検知する．また，感染検知では主に感染したホストのトラヒックデータに着目している．本研究ではその一手法として，ヘッダよりも情報が多く，感染時通信と正常時通信を区別する情報が多く含まれていると考えられる，ペイロードを用いた検知手法について検討した．

本研究ではまず，感染ホストの通信に着目し，そのペイロード情報に基づいて感染検知を行う既存研究を分析した．その結果既存研究では，検知手法の検討がメインで，どの特徴量が正常と感染を区別しやすいかという観点からの検討が十分に行われていないことがわかった．個々の特徴量の有効性を明確にできれば，複数の有効な特徴量を組み合わせることで，より正確に正常と感染を識別できる可能性があると考えられる．

そこで本研究では，CCCDATAset2009, 2010, 2011，D3M2012, 2013，PRACTICE2013 の感染時通信と 2 種類の正常時通信を用い，マルウェアの種類毎（ワーム，トロイの木馬，ファイル感染型ウイルス）における，感染検知の識別に有効な特徴量を評価した．その結果，マルウェアの種類毎に, 特定の ASCII 文字コードと HTTP リクエスト長の True Positive Rate（以下 TPR)・True Negative Rate （以下 TNR）が高く，有効な特徴である可能性があることがわかった．また，4 つの量子化レベルを変化させ，TPR・TNR の平均値が高い上位 15 個の特徴量に着目した結果，TPR95%以上，TNR80%以上の識別率を「高い」とした．具体的な結果としては，TPR のみが高い特徴量としてワームで 3 個，トロイの木馬で 15 個，ファイル感染型ウイルスで 5 個を見いだし，TNR のみが高い特徴量としてワームで 1 個を見いだした（計 20 個の特徴量）．その中でも，TPR・TNR が共に高い特徴量として，ワームでは ASCII 文字コード「i」，ファイル感染型ウイルスでは HTTP リクエスト長が特に有効であることがわかった．

さらに，有効だと判断した 20 個の有効な特徴量を用い，2 つの特徴量の組み合わせ，3 つの特徴量の組み合わせにおいて，最適な組み合わせを評価した．その結果，2 つの特徴量の組み合わせでは，正常時通信と感染時通信を分類する境界線において 3 つ（ASCII 文字コード「e」+「o」（「e」: 99.0% → 100%,「o」: 99.0% → 100%）など），感染時通信を正常時通信と誤検知した感染時通信を分類する境界線において 17 つ（ASCII 文字コード「0」+「i」（「0」: 97.5% → 99.4%,「i」: 98.5% → 99.4%）など）の組み合わせが，7 章で有効だと判断した特徴量の識別率よりも，識別率が 2 つとも高くなった．また，3 つの特徴量の組み合わせでは，感染時通信を正常時通信と誤検知した感染時通信を分類する境界線において，7 つ（ASCII 文字コード「e」+「i」+「o」（「e」: 99.0% → 99.5%,「i」: 98.5% → 99.5%,「o」: 97.0% → 99.5%））の組み合わせが，7 章で有効だと判断した特徴量の識別率よりも，識別率が 3 つとも高くなった．これらを用いることで，より正確に正常と感染を識別できる可能性があることを示した．

本稿では個々の特徴量の有効性，最適な特徴量の組み合わせついて明らかにした．今後は，これらの特徴量，特徴量の組み合わせを用いた検知アルゴリズムについて検討する．

謝辞

本研究を進めるにあたり，終始懇切丁寧な御指導，御助言を賜りました吉浦裕教授に心から深く感謝の意を表します．また，本研究において様々な御助言を頂きました市野将嗣助教，NTT コミュニケーションズ株式会社の畑田充弘様に深く御礼申し上げます．そして，日頃から討論にご参加頂いた吉浦研究室の皆様に深く感謝致します．

関連研究発表

査読付き論文誌

[1] Yusuke Otsuki, Masatsugu Ichino, Soichi Kimura, Mitsuhiro Hatada, Hiroshi Yoshiura, Evaluating payload features for malware infection detection（情報処理学会論文誌 55 巻 2 号掲載予定）

受賞

学生論文賞受賞：
大月優輔，市野将嗣，川元研治，畑田充弘，吉浦裕：マルウェア感染検知のためのトラヒックデータにおけるペイロード情報の特徴量評価，マルウェア対策研究人材育成ワークショップ 2012(MWS2012), pp. 691-698，2012 年 11 月

査読なし国内会議

[1] 大月優輔，市野将嗣，川元研治，畑田充弘，吉浦裕：マルウェア感染検知のためのトラヒックデータにおけるペイロード情報の特徴量評価，マルウェア対策研究人材育成ワークショップ 2012(MWS2012), pp. 691-698，2012 年 11 月
[2] 大月優輔，市野将嗣，川元研治，畑田充弘，吉浦裕：未知マルウェア検知のためのペイロード特徴量の有効性評価，日本セキュリティ・マネジメント学会 2013 (JSSM2013)，2013 年 6 月
[3] 川元研治，市野将嗣，大月優輔，畑田充弘，吉浦裕，甲藤二郎， トラヒックデータを対象とした N-gram 確率密度を用いたマルウェア感染検知手法に関する一検討， 電子情報通信学会ライフインテリジェンスとオフィス情報システム研究会(LOIS)，2013 年 3 月
[4] 市野将嗣，川元研治，大月優輔，畑田充弘，吉浦裕， スコアレベル融合を用いたマルウェア感染検知手法に関する一検討， DICOMO2012 シンポジウム，2012 年 8 月

査読付き国際会議

[1] Masatsugu Ichino, Yusuke Ohtsuki, Mitsuhiro Hatada, Hiroshi Yoshiura, Detection of malware infection using score level fusion with KernelFisher Discriminant Analysis, IEEE Global Conference on ConsumerElectronics, pp.536-537, October 2013.
[2] Kenji Kawamoto, Masatsugu Ichino, Mitsuhiro Hatada, Yusuke Otsuki, Hiroshi Yoshiura, and Jiro Katto, Evaluation of secular changes in statistical features of traffic for the purpose of malware detection, The 1st International Workshop on "Data Mining for Info-Communication Service and its Diffusion" (DMICSiD2012) in SNPD2012 / Springer ,pp.1-11, 2012【selected outstanding paper】

[1]
参考文献

[1] 瀬戸洋一他編著. “情報セキュリティ概論,” 日本工業出版, 2007. ISBN: 978-4819019170.

[2] Mcafee. 忍び寄るマルウェアの脅威/マカフィーのセキュリティ研究レポート. （http://www.mcafee.com/japan/security/rp automotive system security.asp.）

[3] Kaspersky，数字で振り返る 2013 年，入手先 (http://www.kaspersky.co.jp/news?id=207585920)

[4] 情報処理学会：会誌「情報処理」，Vol. 51, No. 3, pp.235-303 (2010).

[5] 二木真明,渡辺勝弘，“マルウェアの基礎知識と検知，防御技術”，2006．入手先 (http://www.soi.wide.ad.jp/class/20060031/slides/27/index_9.html)

[6] 独立行政法人情報処理推進機構，情報セキュリティ白書 2011，pp28-42

[7] 独立行政法人情報処理推進機構（IPA） (http://www.ipa.go.jp/security/txt/2012/11outline.html)

[8] 2013 年サイバー攻撃「三大脅威」，トレンドマイクロ http://www.trendmicro.co.jp/jp/security-intelligence/index.html#br

[9] 金井瑛，“通信の類似性に着目したネットワークインシデント検知手法”， Master’s thesis, 慶應義塾大学大学院，2009．入手先 (http://www.sfc.wide.ad.jp/thesis/2008/master/kanai-mthesis.pdf)

[10] 鵜飼裕司,小林偉昭,中野学 : 脆弱性を利用した標的攻撃のための解析ツール，マルウェア対策研究人材育成ワークショップ 2008（MWS2008）

[11] 溝口誠一郎,Le Malecot Erwan,堀良彰,櫻井幸一:DHCP によって管理されたセグメントに存在する未使用 IP アドレスの監視手法，情報処理学会論文誌，No.41，p55- 60(2008-5)

[12] 藤原将志，寺田真敏，安部哲哉，菊池浩明：マルウェアの感染方式に基づく分類に関する検討，情報処理学会 CSEC 研究報告, Vol. PRMU97, No. 21, pp. 177–182, Mar. 2008.

[13] 森悠樹，市野将嗣，畑田充弘，小松尚久：マルウェア感染時のトラヒック特徴に関する一考察，マルウェア対策研究人材育成ワークショップ 2009（MWS2009）

[14] 宮本貴朗，小島篤博，泉正夫，福永邦雄 : SVM を用いたネットワークトラヒックからの異常検出，電子情報通信学会論文誌，Vol.B，通信 J87-B(4)，pp593-598，2004，入手先

[15] 桑原和也，安藤槙悟，藤原将志，菊池浩明，寺田真敏，趙晋輝：パスシーケンスに基づく Drive-by-Download 攻撃の分類，マルウェア対策研究人材育成ワークショップ 2010（MWS2010）

[16] Wei Lu et.al：Automatic Discovery of Botnet Communities on Large-Scale Communication Networks，the 4th International Symposium on Information， Computer，andCommunications Security ，2009

[17] 山田明，三宅優，田中俊昭，竹森敬祐，“学習データを自動生成する未知攻撃検知システム，” 情報処理学会論文誌，Vol.46，No.8，pp.1947-1958，2005

[18] トレンドマイクロ　セキュリティデータベース，（http://jp.trendmicro.com/jp/home/index.html）

[19] 原島博，画像情報圧縮，オーム社，1991

[20] Linde Y, Buzo A. and Gray R，“An Algorithm for Vector Quantization，” IEEE Trans, Commun, Vol.28 No,1 pp84-95,1980

[21] 総務省・経済産業省連携 ボット対策プロジェクト Cyber Clean Center　サイバークリーンセンター（CCC），入手先 （http://warp.ndl.go.jp/info:ndljp/pid/286615/www.ccc.go.jp/index.html）

[22] 神園雅紀，畑田充弘，寺田真敏，秋山満昭，笠間貴弘，村上純一 ：マルウェア対策のための研究用データセット～MWS Datasets 2013～，マルウェア対策研究人材育成ワークショップ 2013（MWS2013）

[23] 川元 研治，市田 達也，市野 正嗣，畑田 充弘，小松 尚久：マルウェア感染検知のための経年変化を考慮した特徴量評価に関する一考察，マルウェア対策研究人材育成ワークショップ2011（MWS2011）

[24] 秋月俊寛，市野将嗣，甲藤二郎，小松尚久：ロジスティック回帰分析法を用いたトラヒック予測手法に関する一検討，電子情報通信学会技術研究報告，CQ，112(288), pp. 7–12, Nov. 2012.

[25] Rong-En Fan，Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, Chin-Jen Lin ：LIBLINEAR：A Library for Large Linear Classification，入手先（http://www.csie.ntu.edu.tw/~cjlin/papers/liblinear.pdf）

[26] Wireshark：（http://www.wireshark.org/）

付録

付録 A

A.1　Wireshark の使用方法

本研究では, パケットキャプチャリングにおいて Wirwshark を用いている[26]. Wireshark とは，Gerald Combs が開発した network protocol analyzer であり，800 以上のプロトコル解析機能や 85000 以上の display filter が特徴となっている. Wireshark におけるパケットキャプチャリングは，UNIX では libpcap，Windows では WinPcap を用いて行っている.

A.1.1　キャプチャリング手順

キャプチャリング手順において，あるイントラネットにおける，HTTP 取得時の場合について示す．また，キャプチャリング時のスクリーンショットを図 A.1 に示す.

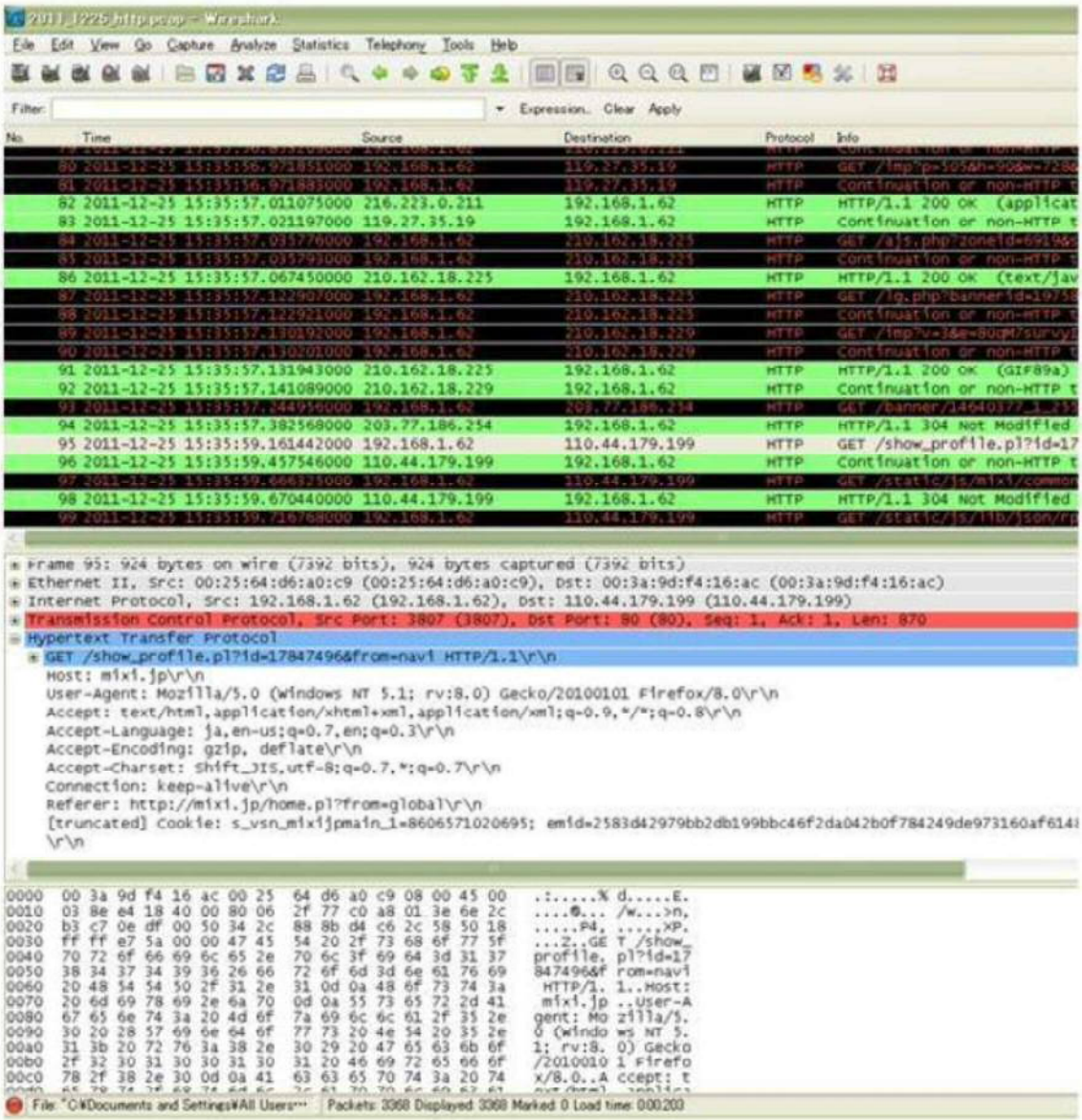

図 A.1 Wireshark によるキャプチャリング時のスクリーンショット

1. Firewall などの常駐ソフトを終了
2. Wireshark を起動し，キャプチャ開始
3. HTTP 通信だけが取得できるようにフィルタリング
4. キャプチャ数 0 の状態で，正常サイトにアクセスを行う
5. 任意時間経過後にキャプチャ終了

A.1.2　ノイズフィルタリング

本研究で用いられたトラヒックデータにおける，ペイロード情報を抽出するためのノイズフィルタリングの一例を以下に示す.

1. 全サービス共通（自身 IP に関わらないパケットの除去）
   ip:addr == 自身の IP アドレス
2. HTTP 通信の処理
   tcp:port == 80&&http
3. パケットキャプチャリング後，任意時間帯のフィルタリング
   frame:time > ”Feb 17; 2012 13：00：00:000000” && frame:time < ”Feb17; 2012 18：00：00:000000”